DEVELOPING MULTI-OBJECTIVE LINEAR PROGRAMMING APPROACHES FOR TRAFFIC SIGNAL OPTIMIZATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MUSTAFA MURAT COŞKUN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JULY 2021

Approval of the thesis:

DEVELOPING MULTI-OBJECTIVE LINEAR PROGRAMMING APPROACHES FOR TRAFFIC SIGNAL OPTIMIZATION

submitted by **MUSTAFA MURAT COŞKUN** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpcılar
Dean, Graduate School of **Natural and Applied Sciences** ______________

Prof. Dr. Mehmet Halit Oğuztüzün
Head of Department, **Computer Engineering** ______________

Prof. Dr. İsmail Hakkı Toroslu
Supervisor, **Computer Engineering, METU** ______________

Dr. Cevat Şener
Co-supervisor, **Computer Engineering, METU** ______________

Examining Committee Members:

Prof. Dr. Cem Evrendilek
Computer Engineering, IEU ______________

Prof. Dr. İsmail Hakkı Toroslu
Computer Engineering, METU ______________

Assist. Prof. Dr. Hakan Yıldız
Computer Engineering, METU ______________

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Mustafa Murat Coşkun

Signature :

ABSTRACT

DEVELOPING MULTI-OBJECTIVE LINEAR PROGRAMMING APPROACHES FOR TRAFFIC SIGNAL OPTIMIZATION

Coşkun, Mustafa Murat
M.S., Department of Computer Engineering
Supervisor: Prof. Dr. İsmail Hakkı Toroslu
Co-Supervisor
: Dr. Cevat Şener

July 2021, 157 pages

In many countries and metropolitan cities, traffic congestion, mainly caused by population growth and the increase in urbanization, has reached a significant level and has become a major problem for residents and decision-makers [1]. Today, although infrastructural strategies such as the construction of underpasses and road widening to alleviate traffic congestion are applied in urban networks, the strategies can also be quite costly and environmentally damaging [2]. Therefore, decision makers allocate large budgets to solve transportation problems and alleviate traffic congestion in urban road networks [3]. In addition, in some cases, infrastructural strategies (road widening, additional lanes, underpasses etc.) can not always be possible due to environmental factors. For this reason, many studies and research have been carried out over the last 40-50 years with the aim of designing and developing effective traffic signal control strategies, which are the most cost-effective and economical solution to the problem of traffic congestion in urban road networks [1]. In this study, various linear programming approaches with different objective functions (fair allocation of residual queue, minimization of total cycle length, minimization of total residual queue, etc.) and fixed-time traffic signal control strategies have been developed, aiming to reduce traffic congestion according to various traffic demands traffic at isolated signalized intersections. In under-saturated conditions, MCLM and CCM linear programming approaches were developed and the green times optimized by the approaches were compared in terms of delay, which is the most important performance criteria using HCM 2000 delay model. In over-saturated conditions, on the other hand, a 2-stage approach has been developed. In the first stage, two different linear programming approaches, MTQLM and MMQLM, were created and the green times of each approach that gave low average vehicle delay according to the traffic scenario were obtained in the first stage with the HCM 2000 delay calculation. In the second stage, the green times obtained in the first scene were optimized to find new candidate green times using HCM 2000 delay model, with the help of neighbor search algorithm (NSM) in a predetermined range. The developed models were evaluated at 3 different types of intersection (T-Type, 4-Legged) with different intersection geometries. The models are compared according to the HCM 2000 average vehicle delay and the reasons for the effects of the models on the delays are also discussed. In addition, in order to measure how generated approaches behave in real-life, the 3rd intersection was also analyzed in the PTV VISSIM microsimulation environment and the delays obtained in the PTV VISSIM environment were compared with HCM 2000.

Keywords: traffic signal optimization, linear programming, delay minimization, vissim

ÖZ

TRAFİK SİNYALİ OPTİMİZASYONU İÇİN ÇOK AMAÇLI DOĞRUSAL PROGRAMLAMA YAKLAŞIMLARININ GELİŞTİRİLMESİ

Coşkun, Mustafa Murat
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. İsmail Hakkı Toroslu
Ortak Tez Yöneticisi
: Dr. Cevat Şener

Temmuz 2021 , 157 sayfa

Birçok ülkede ve büyükşehirlerde, ağırlıklı olarak nüfus artışı ve kentleşmenin artmasının neden olduğu trafik sıkışıklığı önemli boyutlara ulaşmış ve bölge sakinleri ve karar vericiler için büyük bir sorun haline gelmiştir [1]. Günümüzde trafik sıkışıklığını azaltmak için alt geçitlerin inşası ve yol genişletme gibi altyapı stratejileri kentsel ağlarda uygulansa da, stratejiler oldukça maliyetli ve çevreye zarar verici olabilmektedir [2]. Bu nedenle karar vericiler, ulaşım sorunlarını çözmek ve şehir içi yol ağlarındaki trafik sıkışıklığını azaltmak için büyük bütçeler ayırmaktadır [3]. Ayrıca bazı durumlarda çevresel faktörler nedeniyle altyapı stratejileri (yol genişletme, ek şeritler, alt geçitler vb.) her zaman mümkün olamamaktadır. Bu nedenle, kentsel yol ağlarında, trafik sıkışıklığı sorununa en uygun maliyetli ve ekonomik çözüm olan etkin trafik sinyal kontrol stratejilerinin tasarlanması ve geliştirilmesi amacıyla son 40-50 yılda birçok çalışma ve araştırma yapılmıştır [1]. Bu çalışmada, izole sinyalize kavşaklarda çeşitli trafik senaryolarına göre oluşabilecek trafik sıkışıklığını azaltmak amacıyla farklı amaç fonksiyonlarına sahip (artık kuyrukların adil tahsisi, toplam

döngü süresinin minimizasyonu, toplam kalan kuyruğun minimizasyonu vb.) çeşitli doğrusal programlama yaklaşımları ve sabit zamanlı trafik sinyal kontrol stratejileri geliştirilmiştir. Düşük doygunluk koşullarında kullanılmak üzere, MCLM ve CCM doğrusal programlama yaklaşımları geliştirilmiş ve yaklaşımlar tarafından optimize edilen yeşil zamanlar, HCM 2000 gecikme modeli kullanılarak en önemli performans kriteri olan gecikme açısından karşılaştırılmıştır. Aşırı doygun koşullarda ise 2 aşamalı bir yaklaşım geliştirilmiştir. İlk aşamada MTQLM ve MMQLM olmak üzere iki farklı doğrusal programlama yaklaşımı oluşturulmuş ve ilk aşamada HCM 2000 gecikme hesabı ile trafik senaryosuna göre düşük ortalama araç gecikmesi veren yaklaşımın yeşil süreleri elde edilmiştir. İkinci aşamada ise, ilk aşamada elde edilen yeşil zamanlar, önceden belirlenmiş bir aralıkta komşu arama algoritması (NSM) yardımıyla HCM 2000 gecikme modeli kullanılarak yeni aday yeşil zamanları amacıyla optimize edilmiştir. Geliştirilen modeller, farklı kavşak geometrilerine sahip 3 farklı kavşak tipinde (T-Tipi, 4 Bacaklı) değerlendirilmiştir. Modeller, HCM 2000 ortalama araç gecikmesine göre karşılaştırılmış ve modellerin gecikmeler üzerindeki etkilerinin nedenleri de tartışılmıştır. Ayrıca oluşturulan yaklaşımların gerçek hayatta nasıl davrandığını ölçmek için 3. kavşak PTV VISSIM mikrosimülasyon ortamında da analiz edilmiş ve PTV VISSIM ortamında elde edilen gecikmeler HCM 2000 gecikme modelinden elde edilen gecikmeler ile karşılaştırılmıştır.

Anahtar Kelimeler: trafik sinyal optimizasyonu, doğrusal programlama, gecikme minimizasyonu, vissim

To My Family

ACKNOWLEDGMENTS

I wish to express my sincere appreciation to my supervisor, Prof. Dr. İsmail Hakkı Toroslu and my co-supervisor, Dr. Cevat Şener, they convincingly guided and encouraged me to be professional and do the right thing even when the road got tough. Without their persistent support, this thesis study might not have been finalized.

I would also like to thank Metin Barış and Parabol company, who inspired me to choose this master's thesis topic and PTV AG for supporting me to use PTV VISSIM microsimulation environment.

I sincerely thank my friends Ece, Serhat, Engin, Mert, Nurbanu, Özcan, Yusuf, Semih and Volkan who supported me during this stressful process. Their continuous encouragements helped me finalize this master's thesis.

I wish to acknowledge the support and great love of my sisters, Emine Şimşek, Gül Coşkun Ak, Pınar Coşkun, my mother Gülşen Coşkun and my brother-in-laws Bedri Ak, Sezgin Şimşek. They kept me going on, and this study would not have been possible without their support. I also commemorate my father, Hamit Coşkun with mercy for making me come to this day.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

veh	Vehicle
sec	Second
s	Saturation Flow Rate
X	Degree of Saturation (volume-to-capacity ratio) for the lane group
R	Correction Coefficient
c	Capacity of a lane group
Y_c	Total of Critical Flow Ratio
X_c	Degree Of Saturation of an intersection
C	Cycle Length
q	Arrival Rate Per Hour (veh/hour)
λ	Arrival Rate Per Second (veh/sec)
LoS	Level of Service
h	Departure Headway
h_s	Saturation Headway
MCLM	Minimize Cycle Length Method
CCM	Correction Coefficient Method
MMQLM	Minimize Maximum Queue Length Method
MTQLM	Maximum Throughput Method
NSM	NeighbourHood Search Method

CHAPTER 1

INTRODUCTION

In many countries and metropolitan cities, traffic congestion caused by mainly population growth and increase in urbanization has reached significant levels and has become a major problem for urban residents and decision-makers [1]. Therefore, the most primary purpose of traffic engineering is to create more livable cities by reducing the traffic congestion in urban traffic networks as much as possible. However, managing and reducing traffic congestion in cities is still a quite challenging task for decision-makers [10].

Although there is no universally accepted definition of traffic congestion worldwide [11], traffic congestion can be defined as a situation that occurs when the traffic demands on the road networks or the number of vehicles trying to use the road network exceeds the service capacity of the road network. In simple words, when the supply of road networks can not satisfy traffic demands, traffic congestion occurs in urban traffic [12]. In urban road networks or freeways, traffic congestions can be divided into two different categories: ***recurrent traffic congestion*** and ***non-recurrent traffic congestion***. **Recurrent traffic congestion** is a type of traffic congestion that occurs on a normal weekday commute or on an annual period such as vacation times, with no random events such as traffic accidents and bad weather conditions such as sudden heavy hail rain and so on [13]. Therefore, recurrent traffic congestion is a highly expected traffic congestion and the times of occurrence on urban road networks or highways can be highly predicted by decision-makers. Recurrent traffic congestion is generally caused by road network geometries and layouts, inefficient traffic signal control, and fluctuations in traffic flow. **Non-recurrent traffic congestion** in urban networks, on the other hand, is generally caused by random events such as traffic accidents and breakdowns, random bad weather such as sudden heavy hail rain, and so on [13]. Therefore, non-recurrent traffic congestions are generally unexpected traffic congestions and less predictable by decision-makers. As a result, traffic congestion in urban road networks is generally caused by insufficient road capacity, poorly planned road networks, bottlenecks, population growth, the increase in urbanization, traffic accidents, bad weather conditions, the increase in motor vehicles usage, insufficiency of public transportation usage and so on [14].

Traffic congestion in urban road networks results in increased travel times, increased stopped delay, high waiting times and delays at intersections, greater travel costs, frustration in drivers and passengers, increased air pollution (emission of pollutants), and increased number of traffic accidents [2, 10]. Traffic congestion in urban traffic also increases driver stress and frustration, which affects driver behavior and consequently leads to more aggressive behaviors in drivers [15]. Some strategies applied for alleviating traffic congestion in urban road networks can be listed as follows;

- Reducing growth in private vehicle use and extending the use of public transportation vehicles
- Expanding road network infrastructures and intersection capacities
- Providing travel alternatives for drivers and transit vehicles
- Reducing peak-hours traffic demands by spreading the working hours of city residents all day (flexible working hours)
- Reducing physical bottlenecks by road widening, grade-separation of intersecting traffic flows, construction of bridges and underpasses [14]

However, many of these strategies in urban road networks are not always a solution to the traffic congestion problem for some reasons such as environmental factors. For example, road widening at an intersection is not always possible because of the lack of sufficient widening space. Additionally, strategies such as expanding road traffic networks and the construction of new underpasses and bridges are environmentally damaging and quite costly [2]. For this reason, municipalities and decision-makers in our country, as well as in the world, allocate a large part of their annual budgets to alleviate transportation problems and traffic congestion. For instance, Istanbul Metropolitan Municipality allocates approximately **%22** of its annual budget to transportation problems in 2021 [3]. These budgets emphasize the importance that municipalities and decision-makers attach to transportation and transportation problems.

Traffic congestion generally occurs at signalized intersections, which are the bottlenecks of urban road networks. In addition, traffic congestion occurring at an intersection can adversely affect traffic flow and transport in nearby intersections or major streets. For this reason, instead of the strategies for alleviating the traffic congestion problems in urban road networks, especially in signalized intersections, many effective and intelligent traffic signal control strategies have been developed and continue to be developed by researchers for many years. An effective traffic signal control management is a quite cost-effective and economical solution for mitigating traffic congestion, unlike infrastructure-based alleviation strategies [1].

1.1 Motivation and Problem Definition

In light of the aforementioned information, designing and developing an effective traffic signal control strategy is most cost-effective and economical solution to traffic congestion problem in urban traffic network. For this reason, many traffic signal control models and systems have been developed so far by the researchers using different optimization algorithms, and solutions have been sought for traffic congestions occurring at intersections (**see chapter 2**).

Traffic signal control systems developed so far can be categorized into three different categories. **Fixed time traffic signal control** is a traffic control strategy that tries to maximize traffic flow at intersections and its traffic signal settings are generally determined based on historical data at intersections. In this control strategy, traffic signal plan and settings (durations) change only a few time within a day. For this reason, fixed time traffic signal control strategies can not respond to fluctuations in traffic demands. In **traffic-actuated and adaptive traffic control** strategies, traffic signal settings are changed in real-time in line with traffic demands perceived with the help of detectors (sensors) placed at intersections or technologies such as GPS. However, since these strategies use detectors, sensors or any other tools, they are quite cost-expensive because of maintenance and initial deployment of detectors compared to fixed-time traffic control strategies. Some of the traffic signal control models developed so far have focused on isolated intersections, some on coordinated arterial networks, and some on grid networks comprising multiple intersections.

The primary purpose of traffic signal control strategies is to manage traffic flows on intersections or urban road networks in the most efficient way. Therefore, researchers and developers have focused on different objectives in the studies aimed at developing traffic control strategies for an efficient traffic management. These objectives appear in many ways, such as minimizing total number of stops, maximizing traffic flow throughput, minimizing total residual queues, and reducing fuel consumption. In addition, traffic control systems need to consider vehicle and pedestrian safety at intersections.

As a result of the research in the literature, it has been seen that many different optimization techniques and artificial intelligence-based algorithms are used to develop an effective traffic signal control management. The most used approaches are deep learning and reinforcement learning based algorithms, fuzzy logic, genetic algorithm and linear programming approaches. An effective traffic signal control should be cost-effective, fast as possible and applicable to real-life. Traffic flows at intersections should be maximized as much as possible, considering the traffic volumes (demands) on each direction. For this reason, in this thesis, linear programming approach has been preferred for a fast and cost-effective traffic signal control optimization.

Linear programming is a mathematical optimization technique that has been used in many business applications and optimization of real-life problems since the 1950s. In the traffic signal control problem, the use of linear programming in the thesis was deemed appropriate, since values such as green durations, saturation flow rates, arrival rates can be expressed mathematically. Linear programming has been used in many studies for the traffic signal optimization problem in the literature. Based on the research in the literature, some deficiencies of the developed linear programming approaches are as follows;

- Developed linear programming models focused on only single objective function such as minimizing total residue length and maximizing traffic flow throughput

- Some models developed can be applicable only in under-saturated conditions, whereas some models only in the over-saturated conditions

- Average vehicle delay, which is one of the most important performance measure, was not taken into account in the evaluation of the models

- Linear programming models have not been tested in any simulation environment (VISSIM, SUMO etc.)

- The effect of different phase design on delays has not been tested

- In order to test and evaluate the models, it was not generated enough number of traffic scenarios with diverse traffic demands at intersections

- In some directions, unfair residual queues were left by models developed

For this reason, in this thesis, various linear programming approaches with different objective functions have been developed for fixed-time traffic signal control optimization in isolated signalized intersections. In order to overcome deficiencies mentioned in linear programming models used in the literature, some of these objective functions generated are such as minimizing the total residual queue length and fair allocation of residual queues to lane groups in over-saturated conditions, and minimizing cycle length in under-saturated conditions. In the next section, within the scope of thesis, proposed models and solutions are mentioned in detail.

1.2 Proposed Methods and Models

In this thesis study, linear programming approaches are used for optimum fixed-time traffic signal control optimization according to traffic scenarios created according to different saturation conditions and different traffic demands in isolated signalized intersections. Within the scope of the thesis study, linear programming models were developed with different objective functions by considering 2 different saturation conditions (under-saturated and over-saturated conditions) in isolated signalized intersections. While developing the models, as assumed in the linear programming methods developed in the literature, the vehicle arrival pattern was assumed to be constant and uniform.

In signalized intersections, **over-saturated conditions** generally occurs when the total cycle length is not long enough to discharge all vehicles arriving during the red interval. In other words, it means that at the end of each cycle there are some residual queues left in some lane groups to the next cycle. The **under-saturated conditions**,on the other hand, occurs when the total cycle length suffices to discharge all vehicles arriving during the red interval. In other words, in under-saturated conditions, there will be no residual queue left in any lane group for the next cycle.

Within the scope of the thesis, in over-saturated conditions, a 2-stage approach has been developed. In over-saturated conditions, as thought in other studies in the literature, **MTQLM** approach that uses the objective function that try to minimize the total residual queue length was first developed. However, in this approach developed, in some traffic scenarios, especially high traffic volume traffic scenarios, the total residual queues are left only in some lanes. This case points that MTQLM does not act fairly (unfair residual queue allocation) to some lane groups at intersections. For this reason, MMQLM approach has been developed as a second approach in cases of over-saturation conditions. In this approach, fair allocation of total residual queues is aimed, considering the vehicle arrival rates at each direction. In other words, the maximum residual queue left in any lane has been tried to be minimized. In the **first stage** of over-saturated conditions, these two approaches developed are compared in terms of average vehicle delays using the analytical HCM 2000 analytical model and the green times of the method that gives the least average vehicle delay according to the traffic scenario are obtained. In the **second stage**, the neighbor search algorithm (NSM) was developed to further reduce the average vehicle delay. This approach was created to search most appropriate green times neighbour the green times obtained from the first stage within a predetermined range. In this method, while searching most appropriate green times, HCM 2000 delay model is used. At the end of the neighbour search algorithm, the most appropriate green times in terms of delay are obtained within a predetermined range. The neighbor search algorithm has been found to be quite promising in terms of run time and finding the global optimum.

In under-saturated conditions, 2 different approaches have been developed. First, **CCM** approach was developed as an approach that shares the green times according to phases by keeping the total cycle length of the intersection constant. In order to investigate the effect of dynamic cycle length on delays in under-saturated conditions, **MCLM** approach have been developed as the second approach. The objective function in this approach is to minimize the total cycle length, provided that there will be no residual queues for the next cycles. The green times obtained from these two approaches are compared in terms of delays using the HCM 2000 analytical model in under-saturated conditions, and the green times which gives the lowest average vehicle delay are obtained. In the experiments conducted, it has been observed that the green times optimized by CCM approach give lower delay in some scenarios in compared to MCLM, while MCLM approach gives lower delay in some scenarios compared to CCM (see chapter 6).

The flow diagram of the complete algorithm can be seen in the figure 1.1.

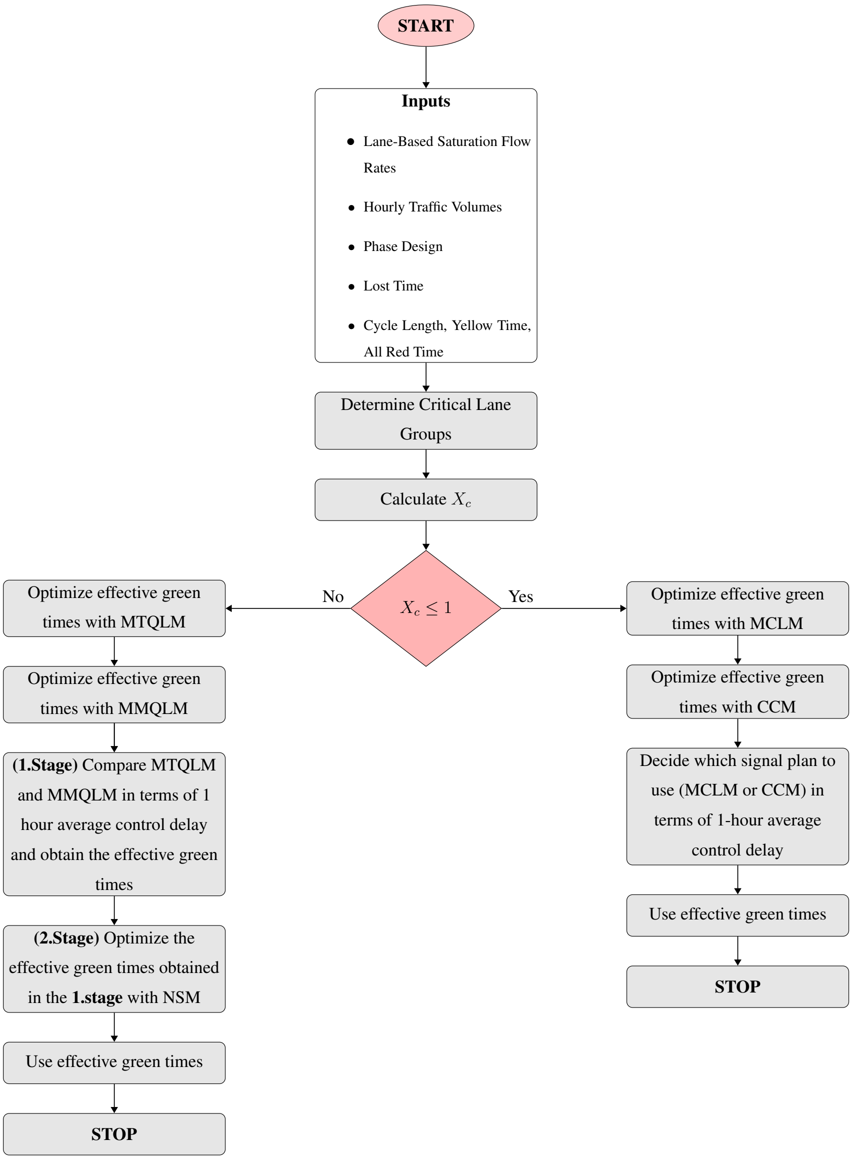


Figure 1.1: The Flow Diagram of Entire Algorithm

1.3 Contributions and Novelties

Within the scope of this thesis, we propose a novel algorithm comprising five approaches with different types of objective can be used both in under-saturated and over-saturated conditions. Our principal contribution is, unlike other linear programming models used in the literature, different objective functions were produced and the advantages and disadvantages of green times obtained according to objective functions in terms of delay, which is one of the most important performance criteria, are discussed.

Our other contributions are as follows;

- Developing different types of objective function according to different traffic scenarios comprising diverse traffic demands
- Using the delay performance criteria besides the queue length criteria while evaluating the models developed
- Testing of linear programming models in PTV VISSIM software, which is a microsimulation environment
- Discussing the effect of different phase designs on delays

1.4 The Outline of the Thesis

This thesis comprises 7 chapters.

In **Chapter 2**, in the literature, we provide information about the studies that approach the traffic signal control problem from different angles and use different technologies and algorithms such as genetic algorithm and reinforcement learning.

Chapter 3 supplies the preliminary information about traffic engineering concepts and traffic signal control strategies. In this chapter, various important concepts such as delay at signalized intersections, which is one of the important performance criteria, saturation flow rate, traffic signal design, phase design are discussed in detail.

Preliminary information about linear programming optimization model, its assumptions and linear solvers used for solving linear programming problems were supplied in **Chapter 4**. In addition, this chapter includes a concrete example of a real-life optimization problem optimized by linear programming.

In **Chapter 5**, different approaches proposed in the thesis study for the traffic signal control problem, linear programming formulizations of these approaches and the advantages and disadvantages of these approaches are explained in detail. In addition, using the sample scenarios created, the effects and reasons of the approaches on the delay are discussed in detail.

In **Chapter 6**, developed models are tested in three different isolated signalized intersections with different intersection geometries (4-legged, T-Type). This chapter includes green times optimized by each developed model and corresponding average vehicle delay results in detail. In addition, in this chapter, the effect of different phase designs on the delay was investigated and the third intersection was calibrated in the PTV VISSIM software, which is a micro-simulation environment. At the end of the chapter, comparison results of the delays obtained by VISSIM software and HCM 2000 model are included.

Chapter 7 includes the conclusion, some recommendations and the possible improvements of the models developed in the thesis study.

Appendix includes numerical formulizations of linear programming models and outputs of linear programming models implemented using Python Pulp package.

CHAPTER 2

RELATED WORK

Today, a good deal of optimization algorithms have been used by researchers for traffic signal control at intersections because of advances in technology, easy access to data, improved analysis of data and increased computing power, and optimization-based traffic signal control has become quite attractive. However, despite these advances in technology, designing effective traffic signal control at intersections is still quite challenging task in terms of defining real-time traffic dynamics at intersections [16]. In many modern cities, many optimization-based traffic signal control systems such as SCATS [17], SCOOT [18], TRANSYT-7F [19], RHODES [20], SYNCHRO [21] are widely used. However, despite this, many optimization-based approaches that approach traffic signal control problem at intersections from different angles have been and continue to be proposed.

Researchers have so far developed many traffic signal control models at intersections, including fixed-time, traffic-actuated, and adaptive traffic signal control. Many approaches such as neural networks [22, 23], fuzzy logic [24, 25], reinforcement learning [26, 27, 28, 29, 30], linear programming [31, 32, 33] and genetic algorithms [1, 34, 35, 36] have been used in the construction of these models.

Researches have mainly focused on isolated intersections, coordinated arterial networks and grid networks. In the proposed models, many optimization objectives such as total queue length, average vehicle delay, average travel time, average waiting time and total network throughput and reduction of fuel emission have been widely used. In researches, real-time data obtained by using different sensors and different technologies were used to observe traffic conditions and traffic dynamics at intersections. In addition, some studies have made use of historical data at intersections. Some of these studies found in the literature are summarized in the rest of this section.

Singh et al. have developed a real-time and intelligent traffic control strategy using genetic algorithms, which they implemented in MATLAB software [1]. By using traffic emulator created using JAVA, traffic conditions are monitored in a certain fixed time interval and these data are transmitted to the genetic algorithm. Using this traffic condition data, the genetic algorithm decides whether to extend the green times of a set of traffic signals, trying to provide near optimum traffic performance. The optimization parameters used by the genetic algorithm are the traffic volume of a road and the importance of it. They verified the efficiency of this system they developed by comparing it with fixed-time and adaptive traffic control systems.

Park et al. [34] have developed a GA-based optimization program to search near-optimal signal timing plan in over-saturated conditions and used a mesoscopic traffic simulator to obtain the fitness (objective) value. The proposed optimization program was compared with the TRANSYT-7F 8.1 on an arterial system comprising 4 intersections created in CORSIM software. 3 different traffic demands volumes (low-medium-high) and 4 performance metrics were used for comparison. Based on the experiment results, GA-based algorithm statistically produced better signal timing plans compared to the TRANSYT-7F in low and high traffic demands, whereas these two programs yielded similar results in medium traffic demands.

Chin et al. have developed a genetic algorithm-based system called GATSTM, which optimizes traffic signal parameters such as offset, green split, phase sequence in order to improve urban network traffic flow and relieve traffic congestion [35]. The system developed aims to minimize lower delays and a better traffic fluency and takes dynamic traffic conditions (adaptive signal control) into consideration. They compared this system with fixed-time traffic signal control and found that GATSTM is better suited to unpredictable traffic network environments than the fixed-time traffic system.

Stevanovic et.al [36] have developed a Vissim-Based genetic algorithm optimization program (VISGAOST) which optimizes traffic signal parameters (cycle length, green split, offset, phase sequence). They tested VISGAOST on 2 different Vissim networks and compared the results with Synchro software. According to the experimental results, using VISGAOST, an improvement of at least 5% was achieved in terms of delays and stops compared to the signal plans produced by Synchro.

Shoufeng et al. [28] used Q-Learning, a reinforcement learning algorithm, to develop an adaptive traffic signal control. They considered the total delay of the intersection as a state and the green time changes as an action. They compared Q-Learning based adaptive traffic signal control developed in this way with the fixed time traffic signal control and concluded that the proposed optimization program gives lesser delay compared to the fixed time traffic signal control.

Liang et al. have developed an optimization program that tries to adjust the traffic light timing creating a deep reinforcement learning model fed with data from different sensors and vehicle networks. In addition, they included components such as dual q-learning and prioritized experience replay to the system in order to increase the performance of the developed model. The reward they used in the system they developed was determined as the total cumulative wait between two cycles. They used SUMO software to test their decision to adjust the traffic light times and the efficiency of the model [27].

Zeng et al. [26] have developed an alternative model using real-time GPS data to control traffic lights at an isolated intersection and they created this model, also called DRQN, by combining the Recurrent Neural Network and the Deep Q-Network. The agent of the model is trained in SUMO software using Q-Learning with experience replay to produce traffic signal control policy. The developed model was compared with the standard Q-Learning method and it was found that it gives a lower average vehicle waiting time compared to standart Q-learning.

Trabia et al. [24] have developed a real-time adaptive signal controller based on two-stage fuzzy logic for isolated intersections. This controller uses vehicle loop detectors at isolated intersections to measure real-time traffic flows and estimates queues on each approach (first stage). Traffic flows and other additional informations obtained from the detectors in a certain fixed interval are used to determine whether a signal phase will be extended (second stage). The researchers compared the model they developed with a traffic-actuated controller on a 4-legged isolated intersection and tested the model's performance.

Kamal et al., unlike previous model predictive control (MPC) based traffic controls, have developed an alternative model predictive control framework that simultaneously adjusts cycle time, green, and offset [37]. They formulated a macroscopic traffic model that perceives traffic dynamics and unexpected changes in traffic conditions on a road network with multiple intersections. They also calibrated the model's parameters with statistical information obtained from a microsimulation software, Aimsun. In a finite horizon, with a given performance index, traffic signals optimized by using Mixed-Integer Linear Programming (MILP). The model developed make the signal lights turn green or red adaptively. The researchers have also been tested and verified the model in a microsimulation software.

Liu et al. [32], in over-saturated conditions, have developed a novel approach using linear programming that tries to maximize the total vehicle throughput by defining traffic arrivals and departures as a smooth continuous function. They defined departures from the intersection as a smooth continuous function instead of commonly used step functions. In this way, the optimum traffic departure rates (the effect) obtained are transformed into green times (the reason) of each phase. With the model developed in this way, they brought a new perspective to linear programming methods in traffic signal control optimization. They evaluated the performance of the model developed with 2 different numerical examples.

Sutarto et al. have proposed a model that minimizes the total queue length and uses the parabolic interpolation technique in MATLAB software by obtaining the traffic arrival and departure rates with a video type vehicle detector for a 2-phase intersection [38]. Since the intersection has 2 phases, they aimed to relieve traffic congestion by optimizing the green time of only a phase. They tested the model on a 2-phase intersection in Jakarta City. Also, they compared the developed model with different cycle lengths and investigated the cycle length that gives the lowest total queue length according to traffic data.

Srisurin et al. [31] have proposed a model that uses a linear programming approach, with minimum green times created using pedestrian safety and intersection geometries, aiming to reduce the total queue length at an isolated intersection. Model developed also aims to find an optimum cycle length to reduce the number of idling (waiting) vehicles. Based on tests performed on a 4-phase intersection in the city of Honolulu, the proposed model was compared with previous fixed-time signal control, and it was found that the proposed model reduced queue lengths by 24.2%-46.1% range.

Coll et al. [33] have proposed an adaptive traffic signal control based on linear programming using traffic data coming from real-time sensors located at intersections. The model aims to reduce the total queue length at intersections. The proposed model has been subjected to performance comparison with previous traffic signal control settings in 2 pilot areas in the city of Buenos Aires. Performance evaluation of the model was carried out in a traffic simulation system and it is stated that the preliminary results are very promising.

Celtek et al. have developed an adaptive traffic signal control that adapts to the momentary traffic conditions at intersections and used swarm-based heuristic optimization algorithms that improve traffic signal control performance [39]. Also, they made a performance evaluation via SUMO traffic simulator. In order to compare the proposed model with the current traffic measurement data, they used real traffic data in Kilis, Turkey. As a result, they concluded particle swarm optimization is suitable and successful in optimizing traffic signal control with real traffic dynamics.

Çakıcı [40] has developed an adaptive traffic signal control that optimizes phase plans and green times of the phases simultaneously. In the proposed model, differential evolution algorithm was benefited to reduce the average vehicle delays using Akçelik Delay Model. In addition, based on the observations at the intersections, it was aimed to dynamically update the phase plans and signal durations of phases in a certain time interval. PTV VISSIM software was used for performance evaluations of the developed model. The model developed in the study was compared with traffic actuated control and fixed-time control in 2 different intersection types (4-Legged and T-type) and it was observed that the developed model performs better in terms of average vehicle delay.

CHAPTER 3

PRELIMINARIES PART 1 : TRAFFIC ENGINEERING

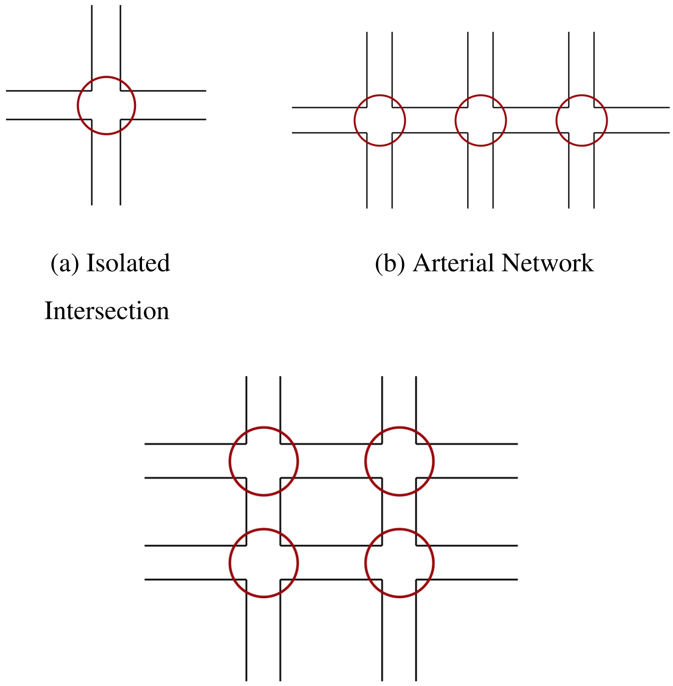
3.1 Intersection Types and Types of Road Network

Intersections are sections of the highway where traffic flows from different directions can cross. In urban areas, vehicle delays, long travel times and traffic congestions are mostly caused by intersections [41].

Intersections are divided into two categories according to whether the traffic flows cross on the same plane (or level) or not. **At-grade intersections** are intersections where traffic flows intersect on the same plane (or level) and rights of way is managed by signalization systems. **Grade-separated intersections** are a type of intersection that aims to reduce vehicle delays and waiting times by separating the possible conflicts of traffic flows with the help of underpasses and ramps. Grade-separated intersections are mostly used in highways as they make vehicle traffic flow faster by eliminating possible traffic flow conflicts [42]. Since the construction and maintenance costs of the grade-separated intersections high, at-grade intersections are generally preferred in urban traffic.

At-grade intersections are generally managed by signalization systems (signalized intersections). However, especially in suburban and rural areas, traffic flows at these intersections are only managed by using a set of traffic signs and no signalization present (unsignalized intersections). In this thesis, only signalized intersections have been studied.

Signalized intersections in urban areas generally form three types of road network: **isolated intersections, arterial network and general (grid) network** [5] (Figure 3.1).



(a) Isolated Intersection

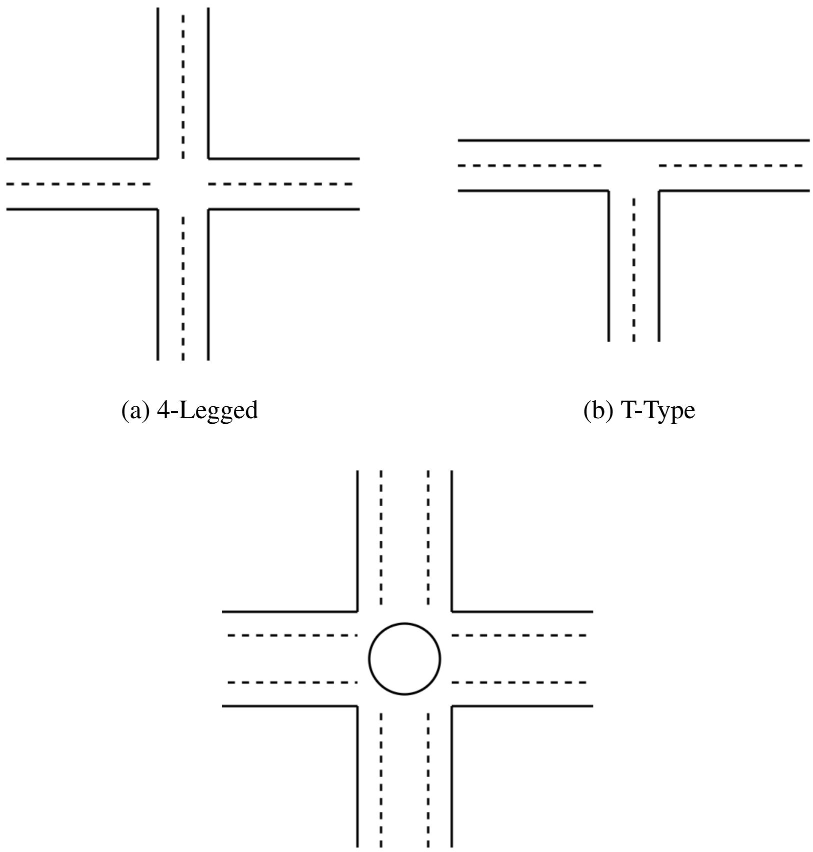
(b) Arterial Network

(c) General (Grid) Network

Figure 3.1: Types of Network in urban areas [5]

HCM defines **isolated intersection** as "an intersection at least 1.6 km from the nearest upstream signalized intersection" [6, p. 55]. An **arterial network** is a road network comprising a series of successive intersections on a road [5]. In arterial networks, many studies have been carried out so far for the coordinated operation of the signalized systems of the intersections such as green wave coordination [43]. All other networks other than these two road networks are called a **general network or grid network** comprising many intersections [5]. In this thesis, only signalized isolated intersections have been studied.

Isolated intersections can have different intersection geometries, traffic environment from city to city and from region to region. Some types (4-Legged, T-Type etc.) of isolated intersections are shown in figure 3.2.



(a) 4-Legged

(b) T-Type

(c) Isolated Intersection with roundabout

Figure 3.2: Isolated Intersection Types and Geometries

3.2 Traffic Signal Control Strategies

In this section, three different traffic signal control (fixed-time, traffic-actuated adaptive) strategies are discussed.

3.2.1 Fixed-Time Signal Control

Fixed-time traffic signal control is an off-line traffic signal control strategy whose traffic signal plan and settings (durations) change a few times within a day. This control strategy uses historical data rather than real-data in determining signal plans and green times at intersections. For this reason, the major disadvantage of this strategy is that it can not respond to random fluctuations (special events, accidents etc.) in traffic flows and dynamics [44]. However, the cost of using this control strategy at intersections is quite low because it does not use any sensors or detectors to obtain instantaneous traffic dynamics. It is also very convenient to use in coordinated arterial networks because the green times and traffic volumes of all intersections in the network are at a predictable level [45].

3.2.2 Traffic-Actuated Signal Control

Traffic-Actuated traffic signal control is a traffic signal control strategy that can respond to real-time traffic demands obtained with the help of detectors placed on the approaches. It can be categorized into two as semi-actuated and fully-actuated traffic signal control.

3.2.2.1 Semi-Actuated Signal Control

In semi-actuated traffic signal control, the detectors are placed only on minor approaches such as side roads where the traffic demand is quite low. Unless there is a traffic demand from these minor approaches, the traffic flows on the major street are given the green time and the right of way continuously. When there is a certain traffic demand on the minor approaches, the phase on the main street is terminated and the minor approaches are given a pre-determined green time and the right of way.

Semi-actuated traffic signal control is more advantageous than fixed-time traffic signal control because it reduces the delay of the major street, especially in coordinated arterial networks with side roads. The most important disadvantage of semi-actuated traffic signal control is that it can cause high delay if the maximum green time and other parameters are not properly set [46].

3.2.2.2 Fully-Actuated Signal Control

In fully-actuated traffic signal control, detectors are placed on all approaches (main and minor roads) and phase green times are automatically determined according to real-time traffic demands.

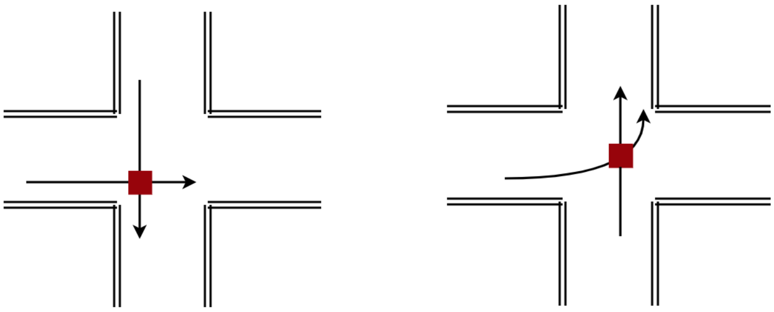
In this traffic signal control strategy, phase green times are determined in a range determined by the minimum and maximum green times. If the traffic demand continues in the currently active phase, the green period is extended until the maximum green time or until the traffic demand on other roads reaches a certain level. Although this signal control strategy is advantageous in reducing delays over fixed-time signal controls, it is disadvantageous compared to fixed-time signal controls because of the maintenance and repair costs of the detectors [46].

3.2.3 Adaptive Signal Control

Adaptive traffic signal control is a traffic signal control strategy that can respond quickly to real-time traffic fluctuations and optimize phase plans and green times to reduce vehicle delay or travel times. Real-time traffic dynamics are obtained using sensors, detectors, camera or GPS data, and signal durations and phase plans are optimized using different optimization algorithms. The adaptive traffic signal control tries to relieve traffic congestion by adapting quickly to sudden accidents and unexpected events. Adaptive traffic signal control is a modified version of full-actuated traffic signal control. Their major difference is that the fully-actuated traffic control determines the cycle length based on the past information, whereas the adaptive traffic signal control predicts the cycle length of the real traffic conditions.

3.2.4 Traffic Flow and Flow Rate

Traffic flow in a road network can be expressed as the total number of vehicles pass through a point in a certain time interval. The total number of vehicles is generally expressed as the hourly traffic volume (demand) or simply the flow rate. Traffic flows can also be expressed in lanes at signalized intersections.



(a) (b)

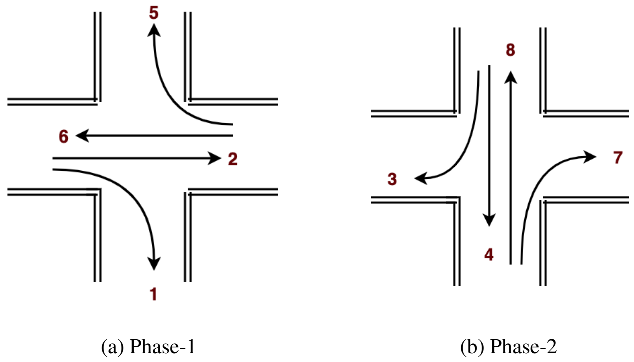
Figure 3.3: Possible conflicts at an intersection

3.3 Lane, Movements and Phase Design

At an intersection, traffic flow in a lane make three types of movements, left and right and straight movements. Depending on different intersection geometry and phase design, some movements or traffic flows share the same lanes and same phase. However, especially at intersections with high traffic volumes, additional lanes are reserved for right movement and left movements while designing intersection geometry.

In traffic engineering, in order to determine when the vehicles coming from different directions will pass through the intersection, phase design should be done first. While designing the phase plan, first, the movements and traffic flows in the directions should be determined by considering the intersection geometry and it should be aimed to prevent potential conflicts at the intersection. Some possible conflicts at intersections can be seen in figure 3.3.

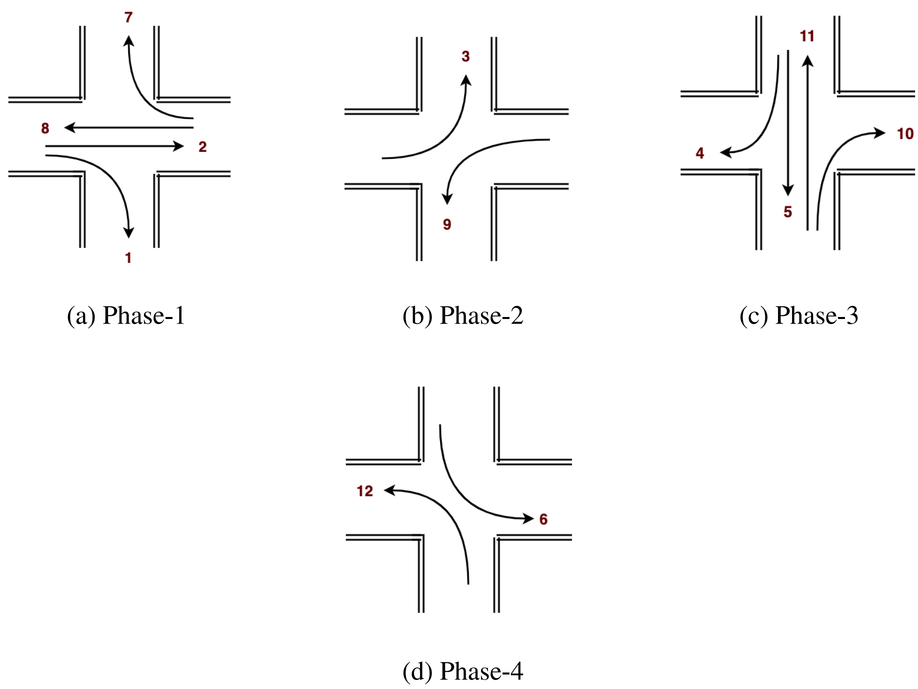
The number of phases may vary according to different intersection types and geometries. In figure 3.4, in an intersection where there are no left turns (movements), it can be seen a sample phase design with **2 phases** and the phases which traffic flows gain right of way.



(a) Phase-1 (b) Phase-2

Figure 3.4: A sample phase design with 2 phases

As seen in figure 3.4, traffic flow 1, 2, 5 and 6 gain the right of way in Phase-1 whereas traffic flow 3,4,7 and 8 gain right of way in Phase-2. In an intersection that includes all types of movements (left, straight and right), a 4-phase signal plan (design) that can be designed can be seen in figure 3.5.



(a) Phase-1 (b) Phase-2 (c) Phase-3

(d) Phase-4

Figure 3.5: A sample phase design with 4 phases

3.4 Traffic Signal Design

After determining a phase design at intersection considering all traffic flows or movements and preventing potential conflicts, traffic signal design can be made.

In traffic lights, three different signal durations are generally applied as actual green time, yellow time and all red time.

3.4.1 Actual Green Time

It is the time that enables vehicles in the traffic flow that gain right of way in a phase to pass through the intersection.

3.4.2 Yellow (Amber) Time, All-Red Time and Intergreen Time

Yellow time or amber time is the time after actual green time and indicates that red time will appear soon. After the end of the actual green time, vehicles can also pass through the intersection during this period [47].

All-Red time in a signal cycle refers to when all traffic lights are all red at the same time. This time does not allow vehicles coming from any direction to pass through the intersection for a certain period, thus preventing potential conflicts or accidents at the intersection and increasing intersection safety [48].

Intergreen time is the time the total time of the yellow times and all-red times in a traffic signal cycle.

3.4.3 Cycle and Cycle Length

The **Traffic Signal Cycle**, or simply **Cycle**, refers to the series of signals where all signal phases gain the right to pass through the intersection. Once the cycle is completed, the starting phase gains the right to pass through the intersection again. **Cycle Length** is the time it takes for a cycle to complete.

A sample 3-phase traffic signal design can be seen in figure 3.6.

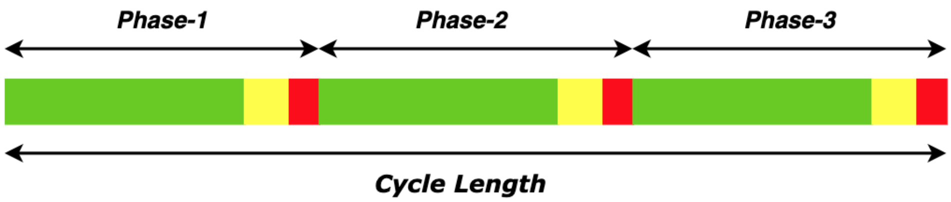


Figure 3.6: A Sample 3-Phase Traffic Signal Design

In the figure, the green areas show the actual green times of the phases, the yellow areas the yellow times after the actual green times, and the red areas show all the red times.

For example, if 10 seconds, 20 seconds, 25 seconds are determined as actual green times for Phase-1, Phase-2 and Phase-3 respectively, and assuming the yellow (amber) time as 3 seconds and the all-red time as 1 second, the cycle length can be calculated as follows;

$$\boxed{\text{C (Cycle Length)} = \text{Sum of Actual Green Times} + \text{Number of Phases} * (\text{Yellow Time} + \text{All-Red Time})} \tag{3.1}$$

According to the equations here, cycle length can be found as **67 seconds** for the 3-phase sample traffic signal design.

3.5 Lane Groups

A lane group is made up with one or multiple lanes and can be used for a separate level of service, queuing, capacity and delay analysis [6]. For example, if straight movements from a direction are provided with three lanes and share the same phase or stop line can be grouped together, and these lanes create a lane group. In the same way, if straight and right movements are provided with the same lane, this can also create a lane group. At some intersections, on the other hand, all three types of movement mentioned before can also create a lane group if they share same lane. However, exclusive right movements and left movements generally create separate lane groups.

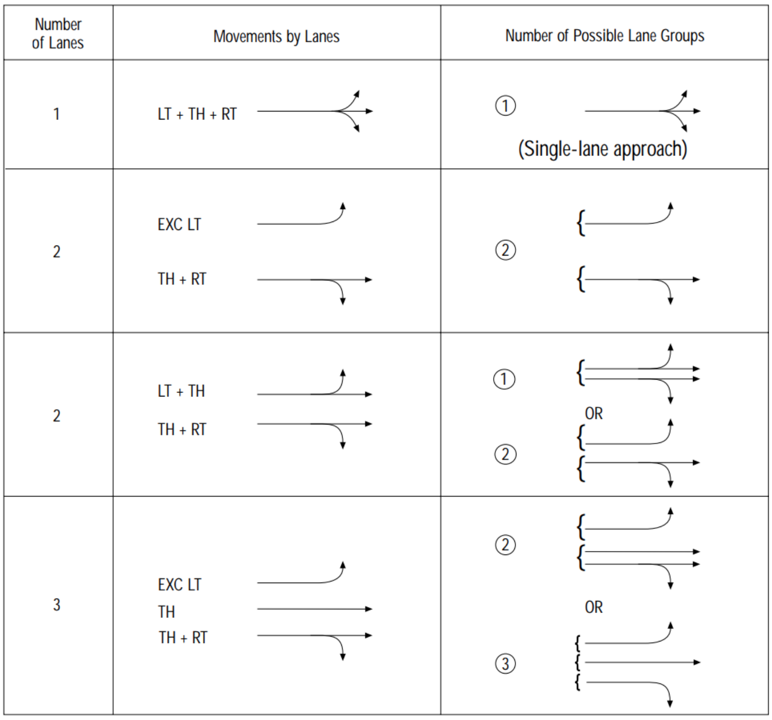


Figure 3.7: Possible Lane Groups For Separate Analysis [6]

3.6 Saturation Flow Rate

One of the most important parameters that should be known in order to design a traffic signal control system effectively at an intersection is the term of **saturation flow rate**. For many years, researchers have investigated which factors will affect saturation flow rate at an intersection and tried to estimate saturation flow rate through measurements and investigations in the field [49, 50, 51].

When the green period starts, vehicles waiting in the queue starts departing from the intersection consecutively. The time between two consecutive vehicles passing the stop line is called as **departure headway** (discharge headway) or simply **headway (h)**. At the beginning of the green period, first vehicle reacts to the green indication and accelerates. Because of the reaction time and initial acceleration, departure

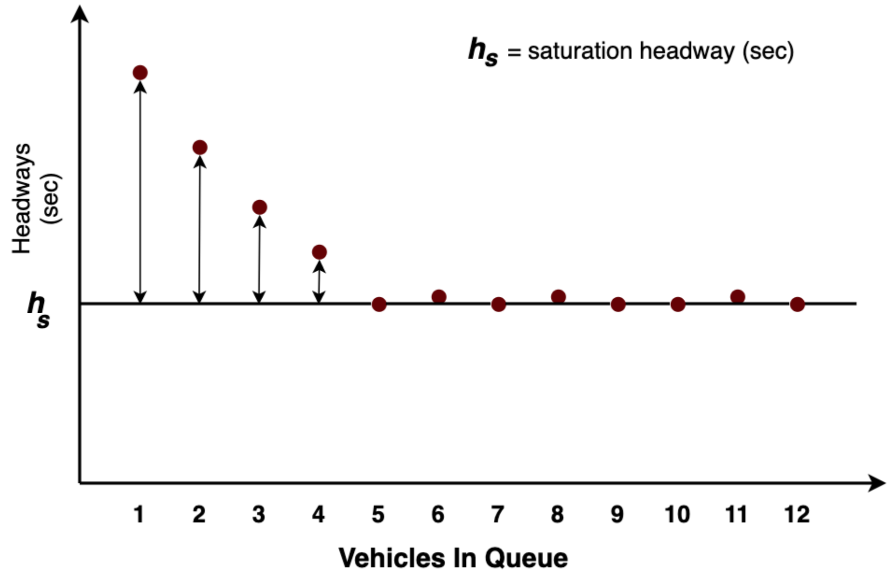


Figure 3.8: Headways and Saturation Headway at intersections [7]

headway between the first and second vehicles is relatively high. Departure headway between the second and third vehicle is relatively smaller than the first departure headway. After a few vehicles, departure headway is fixed to a constant value. This value is called **saturation headway** (h_s).

HCM states saturation flow rate as "The equivalent hourly rate at which previously queued vehicles can traverse an intersection approach under prevailing conditions, assuming that the green signal is available at all times and no lost times are experienced, in vehicles per hour or vehicles per hour per lane" [6, p. 61]. For example, suppose there is a long and sufficient queue in one direction. After the green light turns on, the rate of vehicles entering the intersection is fixed after a certain period. Considering that the green light remains active continuously for 1 hour, saturation flow rate represents the maximum number of vehicles entering the intersection in 1 hour. Saturation flow rate can be calculated by the following formula;

$$\boxed{s = 3600/h_s} \tag{3.2}$$

The saturation flow rate at intersections is affected by many factors. For this reason, it is almost not possible to obtain an exact saturation flow rate at intersection. Therefore, researchers have so far tried to estimates the saturation flow rate with measurements (average saturation headway etc.) obtained from the field in different cities.

The saturation flow rate is generally affected by intersection geometry and traffic environment such (lane width, turning radius etc), different types of vehicles (HGVs, SUVs, motorcyclists or passenger cars) and their compositions, vehicle speeds, rainy or snowy weather conditions and driver behaviors [49, 50, 51, 52, 53]. For example, if the behavior of drivers waiting in the queues is more aggressive when the green period begins, this will lead to a lower safe car-following distance. Therefore, this behavior will also lower the average saturation headway, thus increasing the saturation flow rate. Likewise, in rainy or snowy weather conditions, drivers will increase safe car-following distances, which in turn will lower the saturation flow rate. For these reasons mentioned here, the saturation flow rate varies from city to city in the world, from intersection to intersection.

In many studies in the literature, lane-based saturation flow rate has been assumed to be 1800 veh/hour/lane [49, 40, 24]. Therefore, in this thesis study, according to the literature research, we assumed the lane-based saturation flow rate value as 1800 veh/hour/ lane or 0.5 veh/sec/lane at all intersections where we conducted the experiments. Also, HCM recommended that when vehicle speeds are less than 50 km/h, the saturation flow rate can be taken as 1800 veh/hour/lane [6, p. 172].

3.7 Critical Lane Groups

At intersections, for delay analysis, intersection capacity, queuing and optimization algorithms, lane groups should be first determined. After determining lane groups, critical lane group analysis can be made.

3.8 Effective Green Time and Lost Times

When the green period starts, the vehicles in the queue immediately start to enter the intersection. However, because of the initial acceleration and the first reaction time to green light, a certain period passes for these vehicles to reach saturation flow rate and a lost time occurs. This lost time is called **start-up lost time** in traffic engineering. Likewise, a certain amount of time is lost at the end of the yellow period. This lost time is called **clearance lost time**.

For making queuing and delay analysis at an intersection, **lost time per phase** should be known. **Lost time per phase** is the sum of start-up lost time and clearance lost time. In this thesis study, lost time per phase was assumed to be **2 seconds**.

$$Lost\ Time\ Per\ Phase = Start - Up\ Lost\ Time + Clearance\ Lost\ Time \tag{3.3}$$

The total lost time of the intersection can be calculated by as follows where **N** is number of phases;

$$Total\ Lost\ Time = (\ Lost\ Time\ Per\ Phase * N\ ) + (\ All - Red\ Time * N\ ) \tag{3.4}$$

If a number of phases is 4, all-red time per phase is 1 seconds and lost-time per phase is 2 seconds, then total lost time or simply lost time of the intersection will be 12 seconds.

Effective green time refers to the total time that the traffic flow can pass through the intersection at saturation flow rate. The effective green time for a phase can be calculated as follows;

$$Eff.\ Green\ Time = (\ Actual\ Green + Yellow\ Time\ ) - Lost\ Time\ Per\ Phase \tag{3.5}$$

If actual green time for a phase is 20 seconds, lost-time per phase is 2 seconds and yellow time is 3 seconds, then effective green time of this phase will be 21 seconds.

Finally, **effective red time** for a phase can be calculated by subtracting **effective green time** from cycle length.

Effective green times and lost times can be seen in figure 3.9.

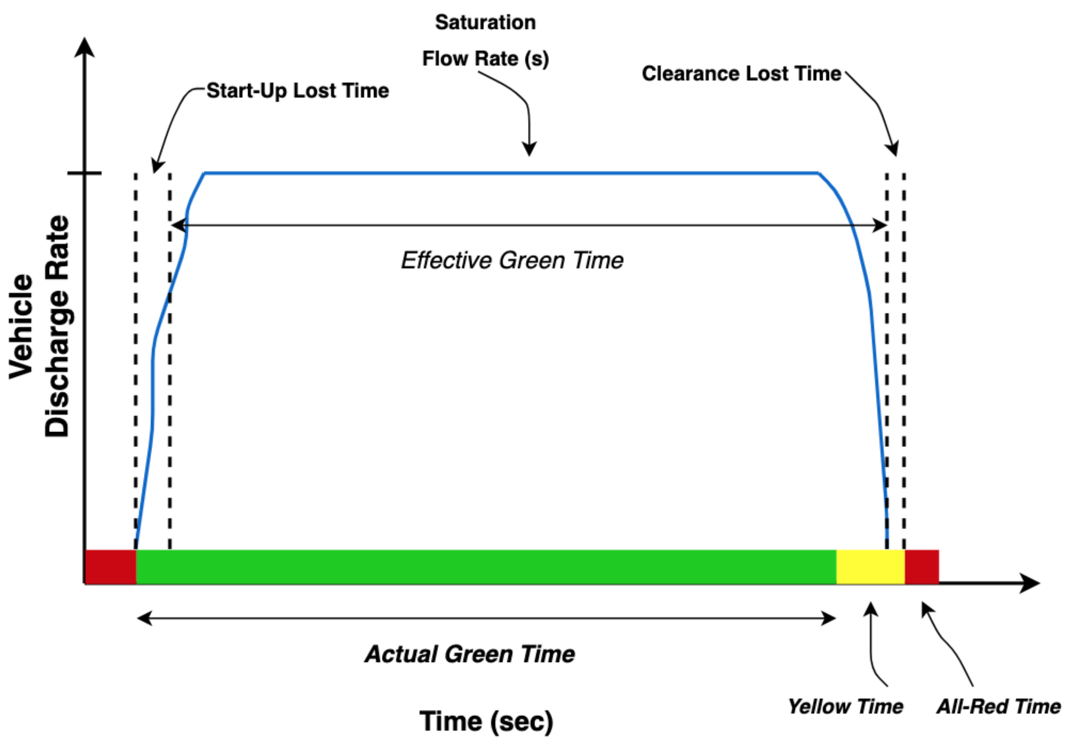


Figure 3.9: Effective Green Times, Lost Times and Saturation Flow Rate

3.9 Capacity

As stated before, the vehicles waiting in the queue leave the intersection according to the saturation flow rate (s) during the effective green time after the green light turns on. For example, let the cycle length is 100 seconds and effective green time for the traffic flow in a lane group is 30 seconds. If the total saturation flow rate is 1800 veh/hour, the maximum number of vehicles that can cross the intersection through that lane group for 1 hour is found as (30/100) * 1800 = 540 vehicles. This value is called as **capacity** in traffic engineering. Capacity can be calculated as follows;

$$\boxed{c = s * (g/C)} \tag{3.6}$$

where:

c= Capacity of the lane group

s= Saturation Flow Rate of the lane group

g= Effective Green Time for the lane group

C= Cycle Length

In equation 3.6, attention should be paid to the saturation flow rate. For example, if there are two lanes in the lane group, saturation flow rate here represents the sum of all saturation flow rates of the lanes.

3.10 Degree of Saturation (Volume-To-Capacity Ratio) of the Lane Group

Another important parameter that should be known in order to understand the queue formation at signalized intersections and to measure vehicle delays correctly is the v/c (volume-capacity) ratio. Traffic congestion reaches serious levels as the hourly traffic volume (q) of a lane group (q, traffic volume) approaches the capacity of that lane group. Therefore, the v / c ratio of each lane group at the intersections should be determined before the delay calculation and queueing calculations are made. The v/c ratio is calculated by dividing the hourly traffic volume (q or v) of the lane group by the capacity of the lane group.

$$\boxed{X = q/c} \tag{3.7}$$

where:

X = degree of saturation of the lane group

q = hourly traffic volume of the lane group

c = capacity of the lane group

Volume-to-Capacity ratio is also known as a degree of saturation of the lane group. Knowing the degree of saturation of a lane group gives us information about the queuing in that lane group. If the degree of saturation of a lane group is greater than 1 assuming uniform arrival pattern, this will lead queue formation at every signal cycle and during the analysis period, queue at the lane group continuously grows. This condition is also called as **over-saturated conditions**. If the degree of saturation of a lane group is less than or equal to 1, this means that the queue accumulated during the cycle in that lane group is cleared at each signal cycle and no residual queue left to the next cycle. This condition is also called as **under-saturated conditions**.

3.11 Deterministic Queueing Model

As mentioned before, assuming that the arrival pattern is **uniform and constant rate** (q), during the effective red time, the vehicles approaching to the intersection forms queues at the approach. At the beginning of the effective green period, queue length reaches the highest value in the cycle Q_{max} (Figure 3.10). When the green light turns on, vehicles waiting in the queue dissipate from the queue with the saturation flow rate during the effective green time. It is important to note that vehicles during the effective green time still arrives to the intersection and continue to be queued at the back of queue. If the traffic demand is less than the capacity of the approach, i.e. v/c ratio or degree of saturation is less than 1, queue is always cleared after each cycle and no residual queue left for the next cycle.

As can be seen from figure 3.10, since the v / c ratio is less than 1, the vehicles accumulated in the queue are always completely cleared after a while within the effective green time (**under-saturated condition**). When there are no vehicles in the queue, vehicles arriving at the lane group pass through the intersection at the arrival rate (q).

The time dependent cumulative arrival function can be expressed as ***A (t)***, and the time dependent cumulative departure function as ***D (t)***.

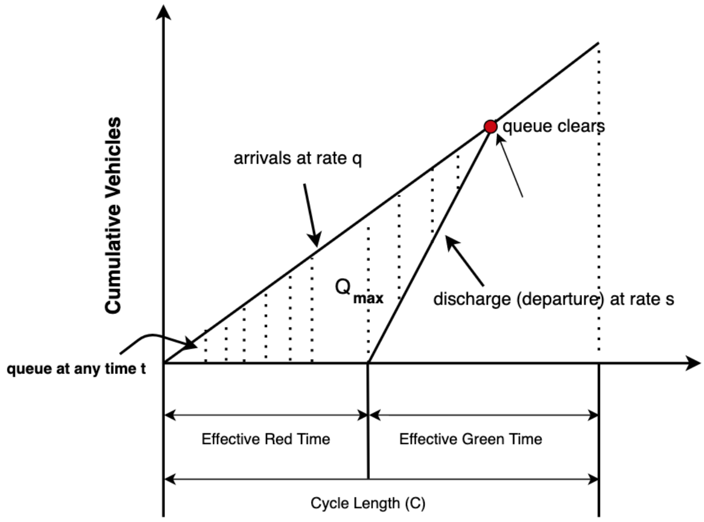


Figure 3.10: Arrival and Departure Pattern - Deterministic Queing (v/c ratio <= 1) [8]

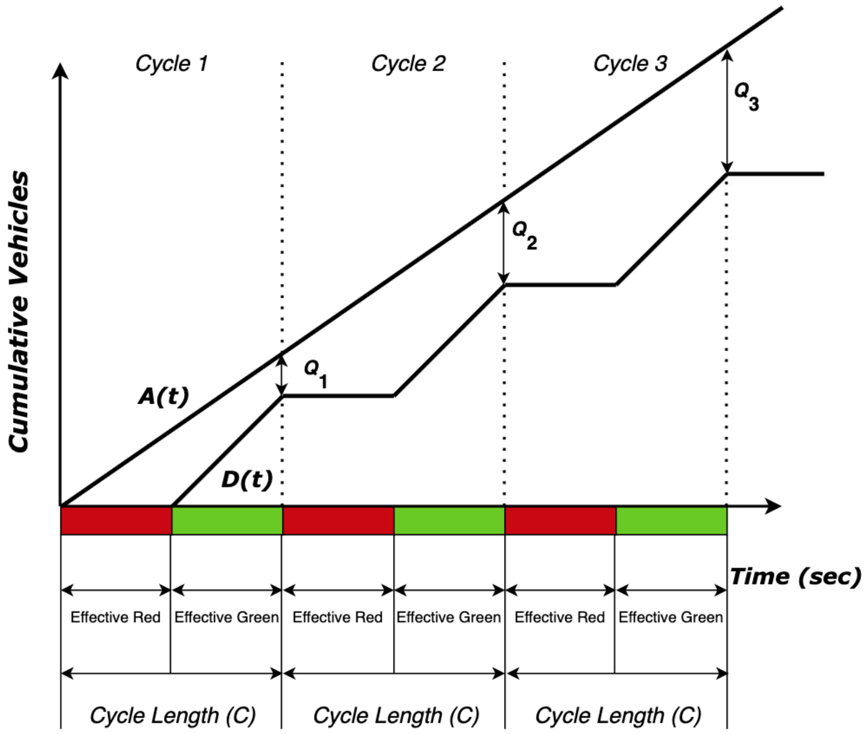


Figure 3.11: Arrival and Departure Pattern - Deterministic Queing - Residual Queues (v/c ratio > 1)

When the v/c ratio or degree of saturation of the lane group is greater than 1, over-saturated condition occurs and the vehicles accumulated in the queue are not cleared completely and some residual queue left for the next cycle. If this condition continues during the analysis period, residual queues for the next cycles continue to grow (Q1, Q2, Q3) (Figure 3.11).

3.12 Critical Lane Groups and Flow Ratio

At intersections, for delay analysis, intersection capacity, queuing and optimization algorithms, lane groups should be first determined. After determining lane groups, critical lane group analysis can be made. In figure 3.12, sample intersection and **six different possible lane groups** can be seen.

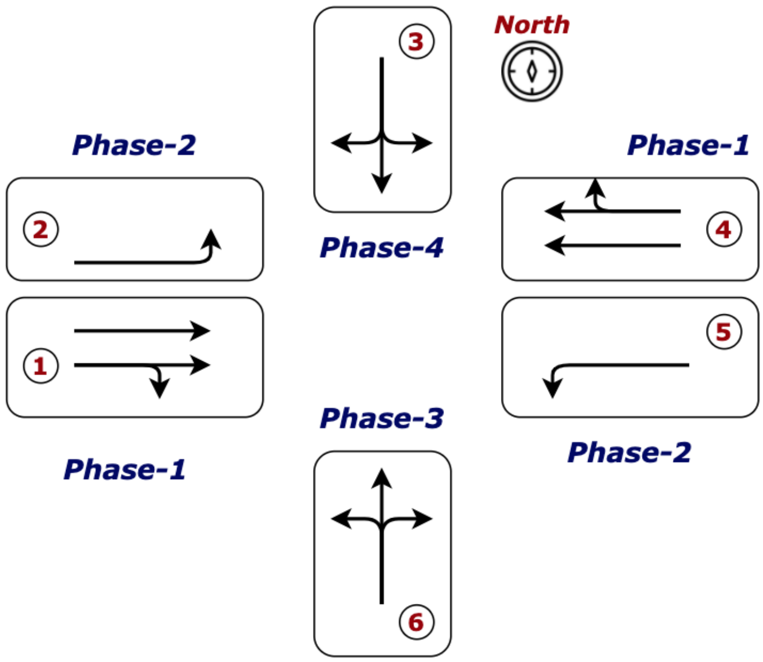


Figure 3.12: A sample intersection and possible lane groups

At this intersection, additional lanes are provided for left turns from both west direction and east direction. These exclusive lanes create separate lane groups, lane group 2 and lane group 5. In lane group 1 and 2, straight-through and right movements share the same lane and same phase. For this reason, those movements can create a lane group. In the north and south direction, 3 types of movement are provided with only one lane. So, they can also create separately lane groups named lane group 3 and 6.

At example intersection in figure 3.12, different lane groups gain right of way in 4 different phases. For example, during the green period of Phase-1, the vehicles on the lane group 1 and lane group 4 can pass through the intersection. In a phase, **critical lane group** is the lane group which needs more green time to discharge its queue. In other words, critical lane group, which has the highest **flow ratio** in a phase. **Flow ratio** of a lane group is simply the ratio of total traffic volume (flow rate) (q or v) over total saturation flow rate (s). Therefore, to find the critical lane groups in all phases, saturation flow rates per lane and traffic volumes should be known firstly. In table 3.1, traffic volumes and saturation flow rates were determined for finding critical lane groups. Saturation flow rates per lane were assumed to be 1800 veh/hour/lane.

Table 3.1: Lane Groups and Flow Ratios

Lane Group No	*Corresponding Phase*	*Traffic Volume(veh/hour)*	*Total Saturation Flow Rate (veh/hour)*	*Flow Ratio*
1	**Phase-1**	**900**	**3600**	**0.25**
2	**Phase-2**	**300**	**1800**	**0.16**
3	**Phase-4**	**400**	**1800**	**0.22**
4	**Phase-1**	**800**	**3600**	**0.22**
5	**Phase-2**	**150**	**1800**	**0.083**
6	**Phase-3**	**250**	**1800**	**0.139**

If we look at the flow ratios of the lane groups, Lane group 1, lane group 2, lane group 6 and lane group 3 are the critical lane groups for the phases Phase-1, Phase-2, Phase-3 and Phase-4, respectively. Other lane groups are called as **non-critical lane groups**.

The sum of flow ratios of critical lane groups, Y_c, in an intersection can be found by equation 3.8.

$$\boxed{\mathbf{Y_c} = \sum (\frac{v}{s})_{ci}} \tag{3.8}$$

where:

v = Total traffic volume of the critical lane group (q)

s = Total saturation flow rate of the critical lane group

Y_c = Sum of flow ratios of critical lane groups

3.12.1 Degree of saturation of the intersection (X_c)

Knowing the degree of saturation of an intersection (X_c) gives information about the queues that may occur in some lane groups. The degree of saturation of an intersection (X_c) can be calculated as follows;

$$\boxed{\mathbf{X_c} = Y_c(\frac{C}{C-L})} \tag{3.9}$$

where:

Y_c = Sum of flow ratios of critical lane groups

C = Cycle Length

L = Total Lost Time

If X_c is greater than 1, this indicates that there will be some residual queue in some lane groups, especially critical lane groups. On the other hand, X_c is less than 1, there will be no residual queue left in any lane groups.

For example, if the flow ratios of critical lane groups are summed at the sample intersection, the Y_c value will be calculated as **0.769**. If cycle length (C) is 100 seconds and the total lost time of the intersection (L) is 10 seconds, X_c value will be approximately **0.85** according to the formula. This value means that there will be no residual queue left for the next cycles in any lane group.

3.13 Performance Criteria : Delay

In traffic engineering, average vehicle delay estimation is an important metric for evaluating the optimization programs that are used for traffic signal timings and measure the intersection performance [9]. For this reason, algorithms and programs that

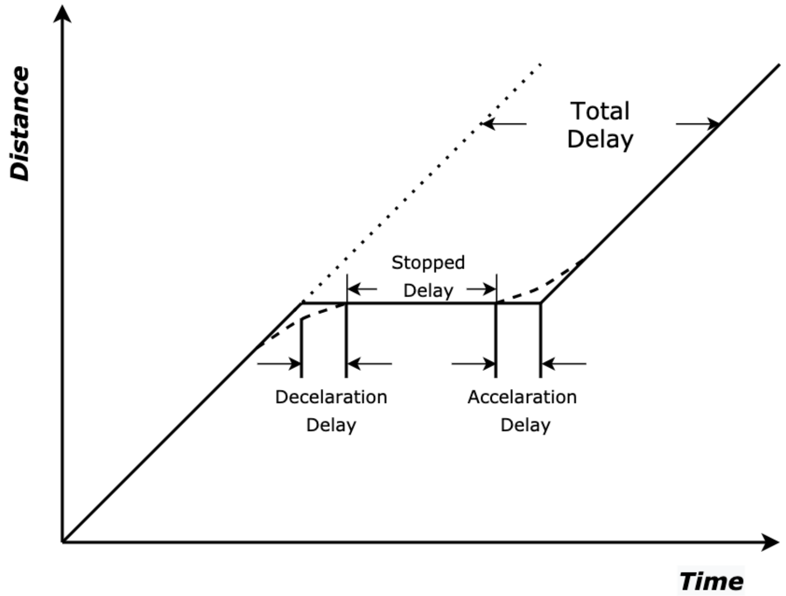


Figure 3.13: Definition of decelaration, stopped and accelaration delays [9]

optimize traffic signals should minimize the average vehicle delay as much as possible for increasing intersection performance.

Delay at signalized intersections can be simply defined as the difference between the normal travel time (no traffic signal, no other vehicles etc.) and the actual travel time caused by the traffic signal, intersection geometry, pedestrians and other vehicles at the intersection.

At signalized intersections, **total delay or control delay** of a vehicle basically comprises three types of delays or components. A vehicle approaching the intersection from an approach lane slows to a stop due to the traffic signal control. This type of delay is named as **decelaration delay**. **Stopped delay** occurs when a vehicle is fully immobilized at the intersection. The more the vehicle waits for the next green indication, the more stopped delay increases. When the green light turns on for that lane or approach, the vehicle starts accelerating to pass through the intersection. In this type of situation, **accelerating delay** occurs until the vehicle reaches its previous or normal speed. Therefore, total delay or control delay experienced by a vehicle is the sum of accelaration, stopped and decelaration delay (Figure 3.13).

Making the delay estimation accurate is very crucial for intersection performance. However, accurate delay analysis, measurement and modeling is quite challenging task. In addition, the materials and workforce required to make accurate measurements in the field make this work quite costly.

For this reason, many simulation softwares have been developed in order to make the delay analysis under the reality and to evaluate the signal durations, phase plans (designs) and intersection geometries in terms of intersection performance. In addition, since the 1960s, many researchers have so far approached delay analysis from different angles and many theoretical and analytical models have been derived [6, 54, 55, 56].

The most famous of these theoretical and analytical models are the Webster [54], HCM [6] and Akçelik methods [55]. These methods give very realistic results in under-saturated conditions. However, these analytical models can not reflect real delays in over-saturated conditions [56].

In this thesis study, using the HCM 2000 analytical model, the proposed linear programming methods were compared in terms of delays and the delay values obtained in the HCM 2000 delay model were also compared with the PTV VISSIM microsimulation model.

3.13.1 HCM 2000 Analytical Delay Model

HCM 2000 [6] is time-dependent delay model that has been widely used by traffic engineers for determining average control delay and level of service (LoS) at signalized approaches for years.

According to HCM 2000 delay model [6, p. 317], average control delay in a lane-group can be calculated by the following equations;

$$\mathbf{d} = \mathbf{d_1}(\mathbf{PF}) + \mathbf{d_2} + \mathbf{d_3} \tag{3.10}$$

$$\mathbf{d_1} = \frac{\mathbf{0.5C(1 - \frac{g}{C})^2}}{\mathbf{1 - [\min(1, X)\frac{g}{C}]}} \tag{3.11}$$

$$\mathbf{d_2} = \mathbf{900T[\ (X - 1) + \sqrt{(X - 1)^2 + \frac{8kIX}{cT}}\ ]} \tag{3.12}$$

where

d = control delay (sec/veh)

d_1 = uniform delay (sec/veh)

d_2 = incremental delay (sec/veh)

d_3 = initial queue delay (sec/veh)

PF = progression adjustment factor

X = volume-to-capacity ratio (v/c or q/c) or degree of saturation for lane-group

C = cycle length(sec)

c = capacity of the lane group(sec)

g = effective green time for the lane group (sec)

T = duration of analysis period (hour)

k = incremental delay adjustment for actuated control

I = incremental delay adjustment for filtering and metering by upstream signals

In this thesis, we have studied on **isolated signalized intersections** which are far from at least 1.6 km from any other upstream signalized intersections [6]. For this reason, as HCM suggested, I, upstream filtering and metering adjustment factor, is included in the formula as 1. Also, we have also studied on optimizing fixed-time signal design in this study. Therefore, k value in the formula is taken as 0.5 for the fixed-time signal control. In the formula, Progression Adjustment Factor is calculated and included in the formula to explain the effect of coordinated traffic signal control.

In isolated intersections, this value is taken as 1. Finally, we assume that there is no initial queue formed before the analysis period, so d3 delay was taken as 0 for all computations.

As mentioned before, if the degree of saturation of a lane group is greater than 1, residual queue continues to grow after each cycle during the analysis period (persistent saturation). On the other hand, if the degree of saturation of a lane group is less than 1, the vehicles accumulated in the queue will always be cleared during the effective green time, so there will no residual queue after each cycle.

According to the HCM delay model, in cases where the degree of saturation is quite low, i.e. not too close to 1, the delay is usually caused by the **uniform delay (d1)** part of the formula. In such cases, the effect of incremental delay (d2) is quite low. However, in cases where the degree of saturation is very close to 1 and greater than 1, the effect of **incremental delay (d2)** is quite high as well as the uniform delay. Moreover, if there is a persistent saturation in a lane group, the incremental delay (d2) will continue to increase as the analysis period gets longer. It is crucial to know this relationship to understand the effect of delays while analyzing intersection performance.

3.13.2 Level of Service (LoS)

To evaluate the performance of an intersection and each lane group, HCM uses level of service (LOS) assessment criterion based on delays that occur. Level of service is an important performance metric to indicate driver discomfort, aggressive behaviour of drivers, fuel consumption, increased waiting times and high travel times [6].

In signalized intersections, the relationship between level of service (LOS) and delays can be seen in table 3.2.

Table 3.2: The relationship between Level of Service (LOS) and Average Control Delay(sec/veh)

Average Control Delay (sec/veh)	*Level Of Service (LOS)*
<= 10	**A**
10-20	**B**
20-35	**C**
35-55	**D**
55-80	**E**
>=80	**F**

As can be seen from table 3.2, the higher average control delays, the worse level of service (**A to F**). Therefore, in a signalized intersection, average control delays of the intersection and each lane group should be reduced as much as possible to improve level of service (LOS).

3.13.3 Numerical Example at a sample intersection : HCM 2000 Delay Model

To explain HCM 2000 delay calculation, a sample intersection was designed comprising 13 different traffic flows (Figure 3.14).

- Traffic flow 1 represents the right movement from west direction to south direction. This flow creates **lane group 1** which can be also named as **west-right** or simply **w-r**.

- Traffic flow 2 and 3 represents the straight movement from west direction to east direction by creating **lane group 2** which can be also named as **west-straight** or simply **w-s**.

- Traffic flow 4 represents the left movement from west direction to north direction. This flow creates **lane group 3** can be also named as **west-left** or simply **w-l**.

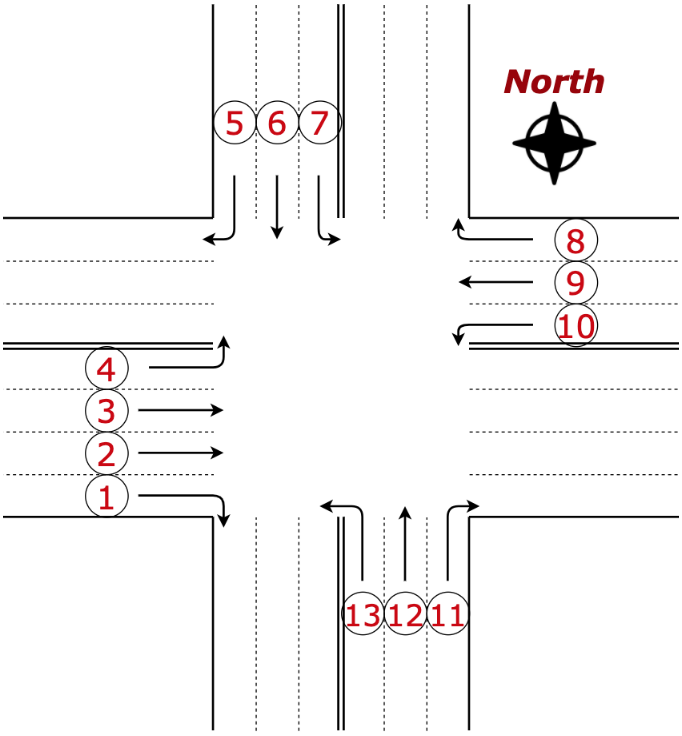


Figure 3.14: Numerical Example (HCM 2000) - A Sample Intersection

- Traffic flow 5 represents the right movement from north direction to west direction. This flow creates **lane group 4** which can be also named as **north-right** or simply **n-r**.

- Traffic flow 6 represents the straight movement from north direction to south direction. This flow creates **lane group 5** which can be also named as **north-straight** or simply **n-s**.

- Traffic flow 7 represents the left movement from north direction to east direction. This flow creates **lane group 6** which can be also named as **north-left** or simply **n-l**.

- Traffic flow 8 represents the right movement from east direction to north direction. This flow creates **lane group 7** which can be also named as **east-right** or simply **e-r**.

- Traffic flow 9 represents the straight movement from east direction to west direction. This flow creates **lane group 8** which can be also named as **east-straight** or simply **e-s**.

- Traffic flow 10 represents the left movement from east direction to south direction. This flow creates **lane group 9** which can be also named as **east-left** or simply **e-l**.

- Traffic flow 11 represents the right movement from south direction to east direction. This flow creates **lane group 10** which can be also named as **south-right** or simply **s-r**.

- Traffic flow 12 represents the straight movement from south direction to north direction. This flow creates **lane group 11** which can be also named as **south-straight** or simply **s-s**.

- Traffic flow 13 represents the left movement from south direction to west direction. This flow creates **lane group 12** which can be also named as **south-left** or simply **s-l**.

Overlapping traffic flows must be carefully designed to increase intersection safety or to prevent accidents at the intersection. In other words, overlapping traffic movements should not be in the same phase. In addition, the intergreen time, which comprises yellow time and all-red time, should be set reasonably. To increase the performance of the intersection and minimize the average delay, a sample phase design created can be seen in figure 3.15.

In this phase design, the right movements gain right of way in 2 different consecutive phases. For example, the right movements from west to south direction gain right of way in Phase-1 and Phase-2.

For this intersection, **120 seconds** cycle length, **3 seconds** yellow time after each actual green time, and **1 second** all-red time was set. Lost time was also assumed to be **2 seconds** per phase. Finally, in the calculations, actual green times were taken as **46, 24, 15 and 19 seconds** for **Phase-1, Phase-2, Phase-3 and Phase-4** respectively. From these assumptions, effective green time per lane group can be calculated (Table 3.3).

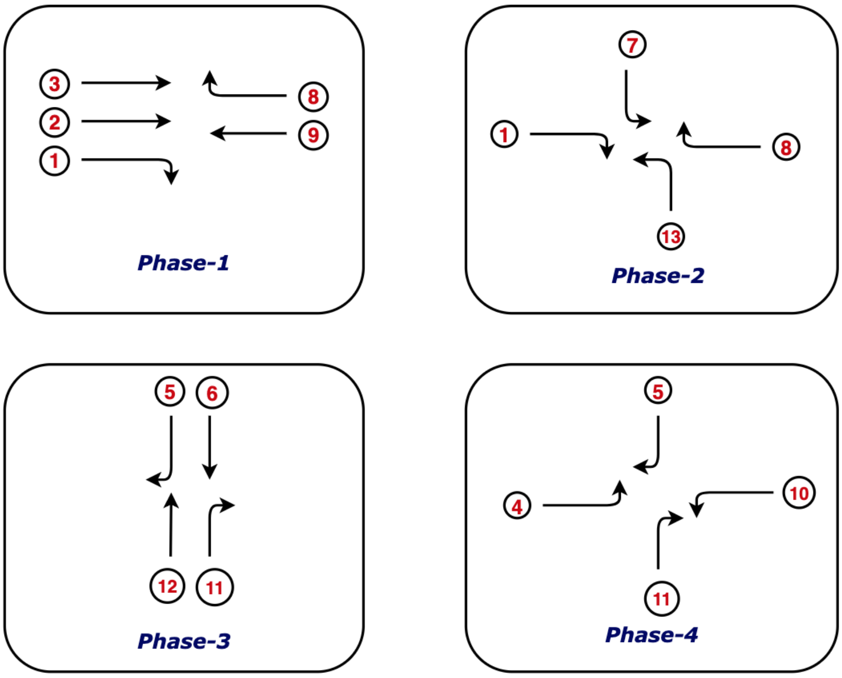


Figure 3.15: Numerical Example (HCM 2000) - A Sample Phase Design

Table 3.3: Numerical Example (HCM 2000) - Effective Green Times Per Lane Group

Lane Group No	*Effective Green Time (g)*	*Yellow Time*	*All-Red Time*	*Cycle Length (C)*
1 (w-r)	**75**	**3**	**1**	**120**
2 (w-s)	**47**	**3**	**1**	**120**
3 (w-l)	**20**	**3**	**1**	**120**
4 (n-r)	**39**	**3**	**1**	**120**
5 (n-s)	**16**	**3**	**1**	**120**
6 (n-l)	**25**	**3**	**1**	**120**
7 (e-r)	**75**	**3**	**1**	**120**
8 (e-s)	**47**	**3**	**1**	**120**
9 (e-l)	**20**	**3**	**1**	**120**
10 (s-r)	**39**	**3**	**1**	**120**
11 (s-s)	**16**	**3**	**1**	**120**
12 (s-l)	**25**	**3**	**1**	**120**

Table 3.4: Numerical Example (HCM 2000) - A Sample Hourly Traffic Volumes Per Direction

	Movements		
	Left	***Straight***	***Right***
West	250 (veh/hour)	600 (veh/hour)	200 (veh/hour)
North	200 (veh/hour)	250 (veh/hour)	150 (veh/hour)
East	150 (veh/hour)	400 (veh/hour)	150 (veh/hour)
South	225 (veh/hour)	200 (veh/hour)	120 (veh/hour)

Saturation flow rate was taken as **1800 veh/hour/lane** for all lanes.

Step 1 : Calculating Saturation Flow Rates Per Lane Group

Total saturation flow rates can be found multiplying number of lanes of the lane group times base saturation flow rates assumed.

- $\mathbf{s_1}$ = 1 * 1800 = 1800 veh/hour , $\mathbf{s_2}$ = 2 * 1800 = 3600 veh/hour
- $\mathbf{s_3}$ = 1 * 1800 = 1800 veh/hour , $\mathbf{s_4}$ = 1 * 1800 = 1800 veh/hour
- $\mathbf{s_5}$ = 1 * 1800 = 1800 veh/hour, $\mathbf{s_6}$ = 1 * 1800 = 1800 veh/hour
- $\mathbf{s_7}$ = 1 * 1800 = 1800 veh/hour , $\mathbf{s_8}$ = 1 * 1800 = 1800 veh/hour
- $\mathbf{s_9}$ = 1 * 1800 = 1800 veh/hour , $\mathbf{s_{10}}$ = 1 * 1800 = 1800 veh/hour
- $\mathbf{s_{11}}$ = 1 * 1800 = 1800 veh/hour, $\mathbf{s_{12}}$ = 1 * 1800 = 1800 veh/hour

Step 2 : Calculating Capacities Per Lane Group

- $\mathbf{c_1}$ = 1800 * (75 / 120) = 1125 veh/hour , $\mathbf{c_2}$ = 3600 * (47 / 120) = 1410 veh/hour
- $\mathbf{c_3}$ = 1800 * (20 / 120) = 300 veh/hour , $\mathbf{c_4}$ = 1800 * (39 / 120) = 585 veh/hour
- $\mathbf{c_5}$ = 1800 * (16 / 120) = 240 veh/hour , $\mathbf{c_6}$ = 1800 * (25 / 120) = 375 veh/hour

- $\mathbf{c_7}$ = 1800 * (75 / 120) = 1125 veh/hour , $\mathbf{c_8}$ = 1800 * (47 / 120) = 705 veh/hour
- $\mathbf{c_9}$ = 1800 * (20 / 120) = 300 veh/hour , $\mathbf{c_{10}}$ = 1800 * (39 / 120) = 585 veh/hour
- $\mathbf{c_{11}}$ = 1800 * (16 / 120) = 240 veh/hour, $\mathbf{c_{12}}$ = 1800 * (25 / 120) = 375 veh/hour

Step 3 : Calculating Degree of Saturation (volume-to-capacity ratio) Per Lane Group

- $\mathbf{X_1}$ = 200 / 1125 = 0.177 , $\mathbf{X_2}$ = 600 / 1410 = 0.425
- $\mathbf{X_3}$ = 250 / 300 = 0.833 , $\mathbf{X_4}$ = 150 / 585 = 0.256
- $\mathbf{X_5}$ = 250 / 240 = 1.04, $\mathbf{X_6}$ = 200 / 375 = 0.533
- $\mathbf{X_7}$ = 150 / 1125 = 0.13 , $\mathbf{X_8}$ = 400 / 705 = 0.567
- $\mathbf{X_9}$ = 150 / 300 = 0.5 , $\mathbf{X_{10}}$ = 120 / 585 = 0.205
- $\mathbf{X_{11}}$ = 200 / 240 = 0.833, $\mathbf{X_{12}}$ = 225 / 375 = 0.6

Step 4 : Calculating Delays Per Lane Group

If the analysis period is set to 15 minutes **(T = 0.25)**, the uniform and incremental delays of each lane group can be calculated.

Uniform Delays of Each Lane Group

- $\mathbf{d_{1-1}}$ = 9.49 sec/veh, $\mathbf{d_{1-2}}$ = 26.65 sec/veh
- $\mathbf{d_{1-3}}$ = 48.39 sec/veh, $\mathbf{d_{1-4}}$ = 29.82 sec/veh
- $\mathbf{d_{1-5}}$ = 52.0 sec/veh, $\mathbf{d_{1-6}}$ = 42.30 sec/veh
- $\mathbf{d_{1-7}}$ = 9.20 sec/veh, $\mathbf{d_{1-8}}$ = 28.55 sec/veh
- $\mathbf{d_{1-9}}$ = 45.45 sec/veh, $\mathbf{d_{1-10}}$ = 29.29 sec/veh
- $\mathbf{d_{1-11}}$ = 50.70 sec/veh, $\mathbf{d_{1-12}}$ = 42.98 sec/veh

Incremental Delays of Each Lane Group

- $\mathbf{d_{2-1}}$ = 0.35 sec/veh, $\mathbf{d_{2-2}}$ = 0.94 sec/veh
- $\mathbf{d_{2-3}}$ = 22.97 sec/veh, $\mathbf{d_{2-4}}$ = 1.05 sec/veh
- $\mathbf{d_{2-5}}$ = 69.40 sec/veh, $\mathbf{d_{2-6}}$ = 5.35 sec/veh
- $\mathbf{d_{2-7}}$ = 0.25 sec/veh, $\mathbf{d_{2-8}}$ = 3.29 sec/veh
- $\mathbf{d_{2-9}}$ = 5.85 sec/veh, $\mathbf{d_{2-10}}$ = 0.79 sec/veh
- $\mathbf{d_{2-11}}$ = 27.45 sec/veh, $\mathbf{d_{2-12}}$ = 6.93 sec/veh

Control Delays of Each Lane Group

Average control delay of each lane group can be calculated by equation 3.10.

- $\mathbf{d_1}$ = 9.83 sec/veh, $\mathbf{d_2}$ = 27.59 sec/veh
- $\mathbf{d_3}$ = 71.35 sec/veh, $\mathbf{d_4}$ = 30.88 sec/veh
- $\mathbf{d_5}$ = 121.40 sec/veh, $\mathbf{d_6}$ = 47.65 sec/veh
- $\mathbf{d_7}$ = 9.45 sec/veh, $\mathbf{d_8}$ = 31.84 sec/veh
- $\mathbf{d_9}$ = 51.30 sec/veh, $\mathbf{d_{10}}$ = 30.08 sec/veh
- $\mathbf{d_{11}}$ = 78.15 sec/veh, $\mathbf{d_{12}}$ = 49.91 sec/veh

Average Control Delay of The Intersection

The average control delay of the intersection is calculated using the average delay of each lane group and the hourly traffic volume of the lane group (equation 3.13).

$$\boxed{\mathbf{d} = \frac{\sum (d_L)(v_L)}{\sum v_L}} \tag{3.13}$$

According to the equation 3.13, average control delay of intersection can be calculated as **d = 46.00 sec / veh.**

Step 5 : Determining Level of Service(LoS) For Each Lane Group

The level of service (LoS) determined according to the delay values of the lane groups can be seen in the table 3.5.

Table 3.5: Numerical Example (HCM 2000) - Level of Service of Each Lane Group

Lane Group No	*Average Control Delay (sec/veh)*	*Level of Service (LOS)*
1 (w-r)	**9.83**	**A**
2 (w-s)	**27.59**	**C**
3 (w-l)	**71.35**	**E**
4 (n-r)	**30.88**	**C**
5 (n-s)	**121.40**	**F**
6 (n-l)	**47.65**	**D**
7 (e-r)	**9.45**	**A**
8 (e-s)	**31.84**	**C**
9 (e-l)	**51.30**	**D**
10 (s-r)	**30.08**	**C**
11 (s-s)	**78.15**	**E**
12 (s-l)	**49.91**	**D**
Intersection	***46.00***	**D**

3.14 Simulation Tools

Although analytical models provide a general insight to evaluate the performance of designed traffic models, the use of simulation softwares is very useful for traffic engineers to observe the change of traffic dynamics, testing certain conditions and parameters, and dynamically analyzing traffic flows in different conditions [57]. For this reason, researchers have used simulation software such as AIMSUN, VISSIM, SUMO to measure and evaluate the real-life performance of the traffic models they have developed so far [37, 36, 27]. The simulation tools can be thought of as an imitation of real-life traffic dynamics and dynamic traffic flows and provide a reliable and comfortable environment for evaluating and calibrating models [57]. For these reasons, traffic simulations are indispensable tools for traffic engineers and transport engineers [58].

Traffic simulation can be divided into 3 categories: microscopic, macroscopic and mesoscopic approach. In the microscopic approach, the individual behavior of vehicles and drivers is designed together with their interactions with other vehicles and pedestrians, and each individual vehicle is treated as a separate object. In the macroscopic approach, traffic flow is considered as a whole. Finally, mesoscopic approach is a hybrid approach that includes both approaches [57]. In recent years, researchers have mostly used SUMO and VISSIM software to evaluate their models and analyze their real-life behavior [57]. For this reason, PTV VISSIM simulation software was used in this thesis study.

3.14.1 PTV VISSIM

PTV VISSIM is a behavior-based and commercial microscopic simulation software that analyzes and optimizes traffic flows, and tests models in real life. It provides detailed reports (vehicle delay, fuel consumption, average queue length, etc.) by analyzing models and traffic flows and has the ability to visualize models in detail. It also allows modeling of suburban, urban and highway applications [58]. Some use cases of PTV VISSIM can be summarized as follows [4];

- Modelling Various Intersection Geometries
- Detailed analysis of many performance criteria such as queue length, fuel consumption, vehicle delay and level of service
- Graphical Depiction of Traffic Flows
- Modelling and Analyzing of Urban Traffic Development Plans
- Investigating, Visualizing Traffic on a Microscopic Level
- Analyzing Traffic-Actuated Signal Control
- Modelling all details for Bus,Subway, Tram, Subway and Parking Operations [4]

Vissim is based on traffic flow model, traffic signal control and their communications (Figure 3.16).

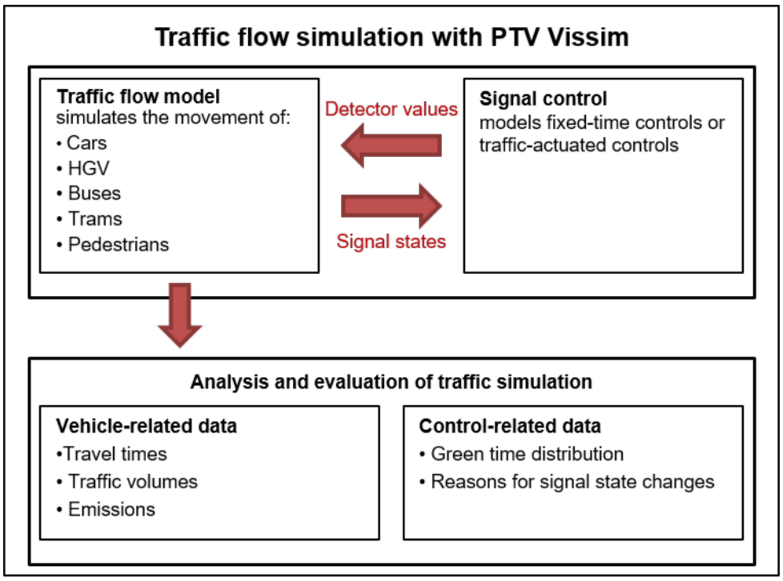


Figure 3.16: Traffic Flow Model and Traffic Signal Control Communication [4]

In VISSIM environment, the saturation flow rate can not be directly given as a value for simulation. Therefore, PTV VISSIM reflects the saturation flow rate using driver behavior and car-following logic. Unlike non-complex models that use deterministic and constant speed-based car-following logic, VISSIM uses the Wiedemann 74 model, which is a psycho-physical car-following model in urban traffic [4]. This car-following model was firstly developed and formulated by Rainer Wiedemann [59].

The Wiedemann 74 car-following model can be summarized as follows; The driver of a faster moving vehicle maintains its free-flow speed until a slower moving vehicle enters the driver's perception threshold. After the slow-moving vehicle enters the perception threshold of the driver of the fast-moving vehicle, the fast-moving driver decelerates and tries to slow down its speed to the slow-moving driver's speed and maintain a safe following distance. After the followed vehicle leaving the driver's perception threshold, the following vehicle speeds up again to reach the free-flow velocity. These situations, which continue as acceleration and deceleration between the following and the followed vehicles, are the general logic behind the Wiedemand 74 model [4].

In order to evaluate the real-life functioning of a designed traffic model and perform its analysis, certain parameters must first be calibrated in VISSIM. Since the default driver behavior parameters of VISSIM can not represent local traffic characteristics and conditions, especially driver behavior and car-following parameters must be calibrated before models can be used in simulation [60].

For this reason, in this thesis, Wiedemann 74 driver-behaviors and car following parameters have been calibrated reducing or increasing the safe following distance to bring the saturation flow rate closer to the desired values.

The following driver-behaviour parameters available in VISSIM;

Table 3.6: Driving Behaviour Parameters - Wiedemann 74 [4]

Parameters	*Description*
Average standstill distance	*(ax): Defines the average desired distance between two cars. Default : 2.0*
Additive part of safety distance	*(bxadd): Value used for the computation of the desired safety distance d. Default : 2.00*
Multiplicative part of safety distance	*(bxmult): Value used for the computation of the desired safety distance d. Default: 3.00*

The desired safety distance d is calculated from:

$$d = ax + (bx_{add} + bx_{mult} * z) + \sqrt{v} \tag{3.14}$$

where:

ax : Standstill Distance

bxadd : Additive part of safety distance

bxmult : Multiplicative part of safety distance

v : Vehicle Speed

z : is a value of range [0.1], which is normally distributed around 0.5 with a standard deviation of 0.15

As can be understood from the formula, the desired safety distance between 2 vehicles can be increased or decreased by decreasing and increasing the parameters of **bxadd** and **bxmult**. In addition, vehicle speeds also have an effect on the desired safe distance.

The higher the desired safety distance, the lower the saturation flow rate. Likewise, reducing the desired safety distance indicates that drivers are moving more aggressively. Therefore, this behavior leads to higher saturation flow rates.

The developers of VISSIM created a scenario to show this situation and by changing the parameters **bxmult** and **bxadd**, they measured the effect of these values on the **saturation flow rate**.

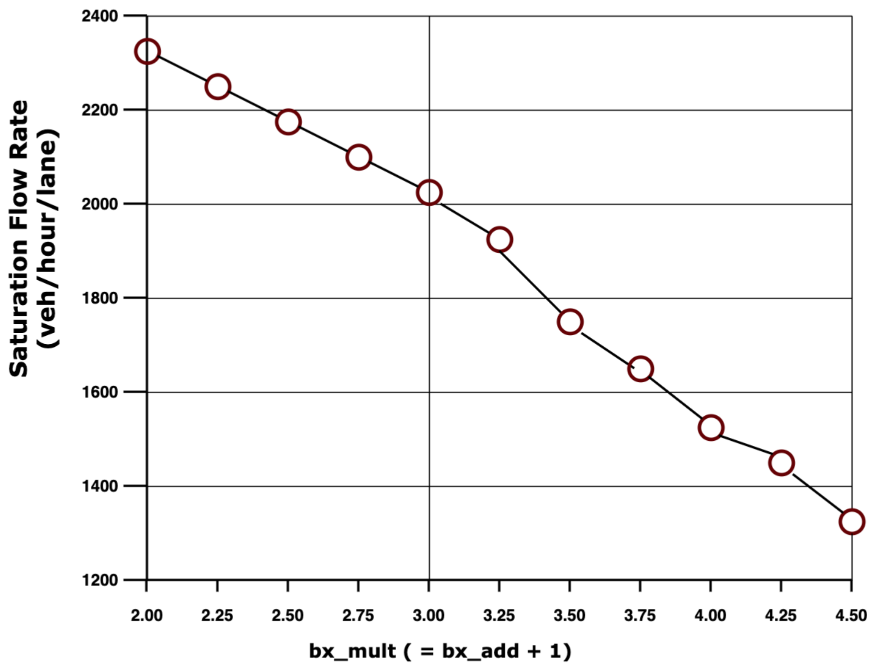


Figure 3.17: The relationship between driving behaviour parameters (bxmult, bxadd) and Saturation Flow Rate

As can be seen from figure 3.17, as the values of the driver behavior parameters increased, the saturation flow rate decreased. Therefore, in this thesis, the saturation flow rates will be tried to be approximated to the desired values by changing these parameters.

CHAPTER 4

PRELIMINARIES PART 2 : LINEAR PROGRAMMING

In its most general definition, optimization can be expressed as planning and finding an optimal solution (decision variables) under certain constraints for the maximization or minimization of a mathematical objective function [61]. Optimization problems have a quite important place in the literature as they are used in many areas and business applications such as agriculture [62], finance [63] and transportation planning [64]. Depending on the application areas, optimization purposes can be goals such as maximizing total profit and reducing energy consumption. In an optimization problem, optimal allocation of limited and scarce resources such as energy, time, workforce is aimed.

In this thesis study, linear programming optimization approach was used for optimizing traffic signal settings and control. In the next section, linear programming approach and its formulization, linear solvers, advantages and limitations will be mentioned.

4.1 Linear Programming

Linear programming is a subfield of mathematical optimization models that provides an optimal allocation of finite and scarce resources in an optimization problem and where the objective function and constraints are defined linearly. The word "linear" means that objective function and constraints of the optimization model are in a linear form. The word "programming" means to the planning of the steps to find the **optimal solution** according to linear constraints and objective function.

In a linear programming model, constraints are in the form of linear equalities and inequalities, and these constraints define the finite and scarce resources in the optimization problem. For example, in the problem of optimal allocation of machines and human resources in the production center of a factory, if the maximum number of products the factory can produce per day is 100, this condition can be included as a constraint in the linear programming model. If the aim of the factory is to maximize daily profit, profit maximization can be determined as the **objective function** of this optimization model. In this optimization problem, the objective function can also be minimization of daily energy consumption. The conclusion that can be drawn from this is that in an optimization problem, the objective function and constraints should be defined carefully and the problem must be translated into linear functions.

A linear program comprises three components: **decision variables** which must be determined optimally, **an objective function** which is defined for making decisions, constraints (equalities and inequalities) which express finite resources and limits of the optimization problem.

A simple linear program can be expressed in **canonical form** as;

$$\boxed{\begin{array}{c} \textit{find a vector} \quad \mathbf{x} \\ \textit{maximizes or minimizes} \quad \mathbf{c^T x} \\ \textit{subject to} \quad \mathbf{Ax \leq b} \\ \textit{and} \quad \mathbf{x \geq 0} \end{array}} \tag{4.1}$$

Here, the components of the vector **x** $(x_1, x_2, ..., x_n)$, are ***decision variables*** that must be determined by a linear solver in order to maximize or minimize the ***objective function***, $\mathbf{c^T x}$. The constraints of the linear program are expressed as $\mathbf{Ax \leq b}$. In addition, all components of the vector **x** must be greater or equal to 0. The vector **x** is a column vector with $\mathbf{nx1}$ dimension where **n** is the number of decision variables. The vector **c** is also a column vector with $\mathbf{nx1}$ dimension. Depending on the number of constraints in the optimization problem, **A** is a matrix with $\mathbf{mxn}$ dimension and vector **b** is a row vector with $\mathbf{1xm}$ dimension.

Linear programming problems can be subclassed according to the value types of the decision variables. If all decision variables are restricted to be an integer in a problem, then the problem is also called as **integer linear programming (ILP)**. Integer linear programming is also called as **binary integer programming (BIP)** if all decision variables are restricted to be integer and with 0 and 1 values. Finally, if some of the decision variables are restricted to be integer and some of other decision variables can take any real value, then the problem is also called as **mixed-integer linear programming (MILP)**.

Linear programming based optimization problems can be solved with different approaches or methods. The most famous one for solving linear programs is the **simplex method** developed by George Dantzig in 1947 [65]. In addition, linear programming problems can be solved using many approaches, such as the **interior-point, branch-and-bound method, graphical method**. In this thesis study, linear programs were written with **Python programming language and Pulp package** which is an open-source and efficient LP modeler. Pulp package uses COIN-OR Branch and Cut Solver (CBC) which is written in C++ by default. This solver can be replaced with another solver as needed.

4.1.1 A Simple Linear Program - Graphical Method

In this subsection, a simple linear program are solved by a graphical method.

A simple linear program can be expressed in **standard form** as;

$$\boxed{\begin{array}{c} \textbf{\textit{maximize}}\ \mathbf{z} = (\ \mathbf{2} * \mathbf{x} + \mathbf{3} * \mathbf{y}\ ) \\ \text{s. t.} \\ \mathbf{4} * \mathbf{y} - \mathbf{3} * \mathbf{x} \leq \mathbf{30}\ ,\ \ (\mathbf{1.\ Constraint}) \\ \mathbf{x} - \mathbf{2} * \mathbf{y} \leq \mathbf{4}\ ,\ \ (\mathbf{2.\ Constraint}) \\ \mathbf{x} + \mathbf{y} \leq \mathbf{10}\ ,\ \ (\mathbf{3.\ Constraint}) \\ \mathbf{x}, \mathbf{y} \geq \mathbf{0}\ ,\ \ (\mathbf{4.\ Constraint}) \end{array}} \tag{4.2}$$

In this linear program, x and y are **decision variables** and z is the **objective function** that needs to be maximized. The x and y variables that maximize the objective function must also satisfy **four different constraints**. In this problem, all constraints are

inequality constraints.

The representation of all constraints which are simple linear functions in the coordinate plane can be seen in figure 4.1.

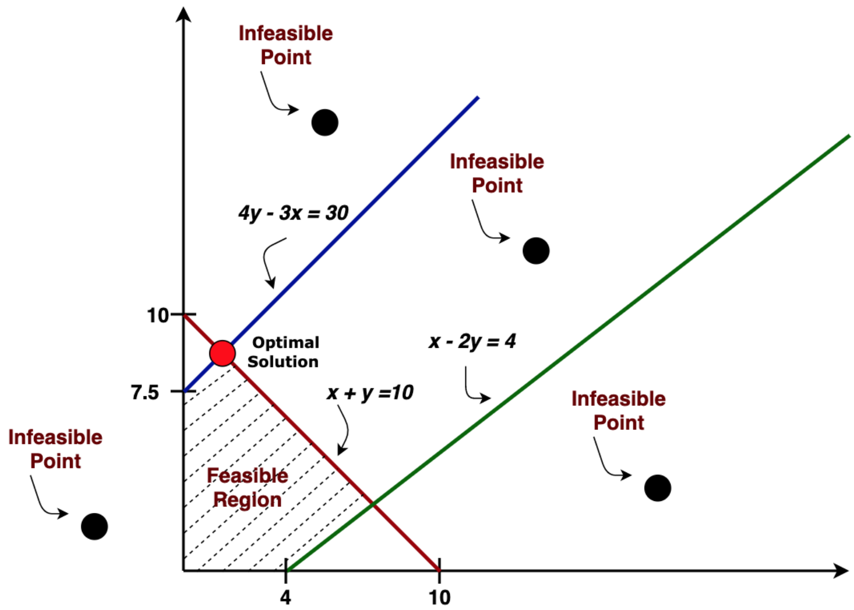


Figure 4.1: Linear Program - Graphical Method

In the figure, four different **infeasible points** was shown. These points can not satisfy four different constraints and thus they can not be used as a solution. All the constraints of the program have created a dashed convex area in the figure. This area is also called the **feasible region**, and every combination of x and y in this area is a potential solution to the problem. No point outside this area can be a potential solution.

In this program, z is a linear objective function to be maximized. According to the graphical method, (x, y) pair which maximizes the objective function should on the **vertexes of the feasible region**. Therefore, in this method, the optimal values according to the constraints will be found as **1.43** for the **decision variable x** and **8.57** for the **decision variable y**. These values are the values that maximize the objective function to **28.57**.

In a linear programming model, constraints and objective function should be determined carefully. For example, if ***x + y <= 10 constraint*** is removed from this program, an infinite feasible region will be occurred. In this case, the problem will have an infinite number of optimal solutions. In such a case, the linear program is called as an **unbounded** linear program. Likewise, if a constraint, such as ***y >= 9***, is added to the program, then no feasible region will occur. In this case, linear program is called as **infeasible** linear program.

4.1.2 A Real-World Optimization Problem

In this subsection, resource allocation and profit maximization optimization problem in bicycle production of a bicycle factory are solved using linear programming approach.

A bicycle factory produces three different types of bicycle, **City Bike, Road Bike and Mountain Bike**. The number of products produced for city bikes is named as **Type1**, for road bikes **Type2**, and for mountain bikes **Type3**. The aim of the factory is to maximize its **daily profit** by using its resources effectively in bicycle production. For this purpose, the factory wants to find out how much of which bike it needs to produce per day (*Type1, Type2, Type3*).

In the production of city bikes, 10 kg of aluminum is consumed per bicycle. 8 kilograms of aluminum and 5 kilograms of plastic are used while producing road bikes. In the production of mountain bikes, 12 kg of aluminum and 6 kg of plastic are consumed. Finally, per unit profit for city bike is **$100**, road bike is **$200**, and mountain bike is **$150**.

The factory should consider the following conditions in bicycle production;

- The maximum number of bicycles the factory can produce per day is 40 (***Factory Capacity Constraint***).
- The maximum amount of aluminum that can be consumed per day is 600 kilos and the maximum amount of plastic that can be consumed per day is 150 kilos (***Resource Constraint***).

- According to customer requests, at least 5 pieces of each type of bicycle must be produced per day (***Customer Request Constraint***).

$$
\boxed{
\begin{array}{c}
\textbf{maximize}\; profit = (\, 100 * Type1 + 200 * Type2 + 150 * Type3 \,) \\
\text{s. t.} \\
Type1 + Type2 + Type3 \leq 40\ , \quad (Factory\ Capacity\ Constraint) \\
10 *\ Type1 + 8 *\ Type2 + 12 *\ Type3 \leq 600\ , \quad (Aluminum\ Constraint) \\
5 *\ Type2 + 6 *\ Type3 \leq 150\ , \quad (Plastic\ Constraint) \\
Type1,\ Type2,\ Type3 \geq 5\ , \quad (C.\ Request\ Constraint)
\end{array}
}
\tag{4.3}
$$

If we solve this real-world problem by using the ***Python Pulp package*** with the constraints and objective function, the daily production quantities that maximize the daily profit will be **11** for city bikes, **24** for road bikes and **5** for mountain bikes. According to the constraints, the maximum profit per day will be **$6650**.

The output produced by **Pulp package** can be seen in the figure **4.2**.

```
Profit_Maximization:
MAXIMIZE
100*Type1 + 200*Type2 + 150*Type3 + 0
SUBJECT TO
Customer_Request_Constraint___Type1: Type1 >= 5

Customer_Request_Constraint___Type2: Type2 >= 5

Customer_Request_Constraint___Type3: Type3 >= 5

Factory_Capacity_Constraint: Type1 + Type2 + Type3 <= 40

Aluminum_Resource_Constraint: 10 Type1 + 8 Type2 + 12 Type3 <= 600

Plastic_Resource_Constraint: 5 Type2 + 6 Type3 <= 150

VARIABLES
Type1 free Integer
Type2 free Integer
Type3 free Integer

Maximum Profit ($) :  6650.0
Type 1 (City Bike) : 11.0 Type 2(Road Bike) : 24.0 Type 3(Mountain Bike) : 5.0
```

Figure 4.2: Real-World Problem (Bicycle Production) - Pulp Package Output

4.1.3 Assumptions of Linear Programming

To determine whether the linear programming approach can be applied to an optimization problem, the assumptions of linear programming must be known [66].

1. **Proportionality :** The contribution of each decision variable to the objective function and linear contraints constraints is proportional to its value.

2. **Additivity :** Objective function and linear constraints are formed by the sum of the individual contributions of each decision variable. In other words, the contribution of decision variables is independent of each other.

3. **Divisibility:** Decision variables are not restricted to be integer values only, and can take any real value such as fractional values.

4. **Certainty:** The coefficients inside the objective function and constraints such as price of a product are known with certainty and known constant. In many real applications, certainty is not always possible. Linear programming approaches are an approach used to optimally determine future actions. Therefore, coefficients which are in the objective function and constraints can be determined and assumed by estimation of future conditions. However, in this case, there would be some uncertainty within the model. [66].

4.1.4 Advantages and Limitations of Linear Programming

Although linear programming is a powerful mathematical optimization approach, it has some advantages and some limitations.

Advantages of Linear Programming

- Linear programming can be used effectively in many wide range of complex business problems.

- Linear programming models is simple and easy understanding.

- Linear programming models provide optimum use and allocation of productive but constrained resources. It also provides insight to decision makers on how to use resources effectively [67].
- Linear programming helps businesses make decisions in a wide variety of conditions and is useful for seeing the differences between decisions for those conditions and thus improves quality of decisions.

Limitations of Linear Programming

- In some business applications and areas, the objective function and the constraints of the system cannot be defined as linear functions.
- Clear determination of a quantitative objective function and constraints associated with the objective function may not always be possible in some real-life problems [67].
- In a real-life problem, the factors (coefficients) that need to be included in the constraints are not always measurable. For example, the productivity of a worker in a factory may drop for psychological or physiological reasons or increase for other reasons. However, this productivity can not be measured precisely [67].
- Linear programming models do not consider effect of time and uncertainty. For this reason, some values and coefficients have to be assumed by estimation according to changing and future conditions.

CHAPTER 5

PROPOSED SOLUTIONS AND MODELS

5.1 Limitations of Linear Programming Approaches for Delay Minimization

In this thesis study, the objective of all generated methods is to assign optimized effective green times of the phases for relieving traffic congestion of the intersection and minimizing delay in terms of HCM 2000. Although linear programming is good and fast optimization technique for solving problems, which can be represented as a linear constraint and linear objective functions, the purpose of delay minimization at an intersection can not be exactly represented by linear constraints. If we look at the incremental delay calculation formula of the HCM 2000 (Figure 5.1) delay model mentioned earlier, the capacity of a lane group is in the denominator part of the formula. Moreover, the capacity formula includes the phase effective green time inside, which is decision variable in all linear programming approaches developed.

If we select the objective function as the average control delay suggested by HCM 2000, the problem becomes non-linear form. For this reason, linear programming

$$\mathbf{d_2 = 900T[\ (X-1) + \sqrt{(X-1)^2 + \frac{8kIX}{cT}}\ ]}$$

capacity of a lane group

Figure 5.1: Incremental delay formula of HCM 2000 that makes the problem non-linear

can not solve the problem with **delay minimization objective**.

At an intersection, optimizing cycle length is also very critical to minimize average control delay. The total cycle length should be neither too short nor too long [44]. If the cycle length is too short, the effective green times will be too short to discharge the vehicles waiting in the queue. Moreover, the ratio of total lost time in a phase over total cycle length is higher and therefore this situation may increase the average control delay, contrary to the expectations. If the cycle length is too long, the vehicles in the queue need to wait more time for the next green period to pass through the intersection. This situation may also increase the average control delay at the intersection. For the reasons described here, optimizing the total cycle length can be also a reasonable objective for optimization program. However, the problem will also be non-linear form with this objective.

5.2 Relationship Between Residual Queue and Critical V/C Ratio

At an intersection, during the effective red period, the vehicles arrives to the intersection, accumulate and form a queue until the effective green period. Even during the effective green period, vehicles continue to arrive at the intersection and join the back of the queue. Residual queue occurs in a lane or lane group in a condition in which an effective green time assigned to that lane is not long enough to discharge all the queue at the end of effective green time. According to the deterministic queuing model mentioned before, if volume-to-capacity ratio of a lane or lane group is greater than 1, the residual queue continues to grow during the analysis period.

At an intersection, first, critical lane groups per phase has to be determined to see whether there will be residual queue (remaining queue) in some lane groups or not. If none of the critical lane groups has residual queue to the next cycle, other non-critical lane groups also have no residual queue but vice versa is not always correct. After determining critical lane groups, the degree of saturation (volume-to-capacity ratio) of the intersection (**Xc**) has to be calculated. If **Xc** is greater than 1, this situation tells us that some critical lane groups will have residual queue to the next cycle (see Chapter 3 for details).

5.3 Proposed Linear Programming Approaches For Traffic Signal Optimization

In this thesis, for different traffic scenarios and different types of intersection, 5 different methods were generated. If **Xc** is greater than 1, MTQLM, MMQLM and NSM were used and compared in terms of HCM 2000 delay model. If **Xc** is less than or equal to 1, MCLM and CCM were used and compared. Superficial flow chart for this workflow can be seen in figure 5.2.

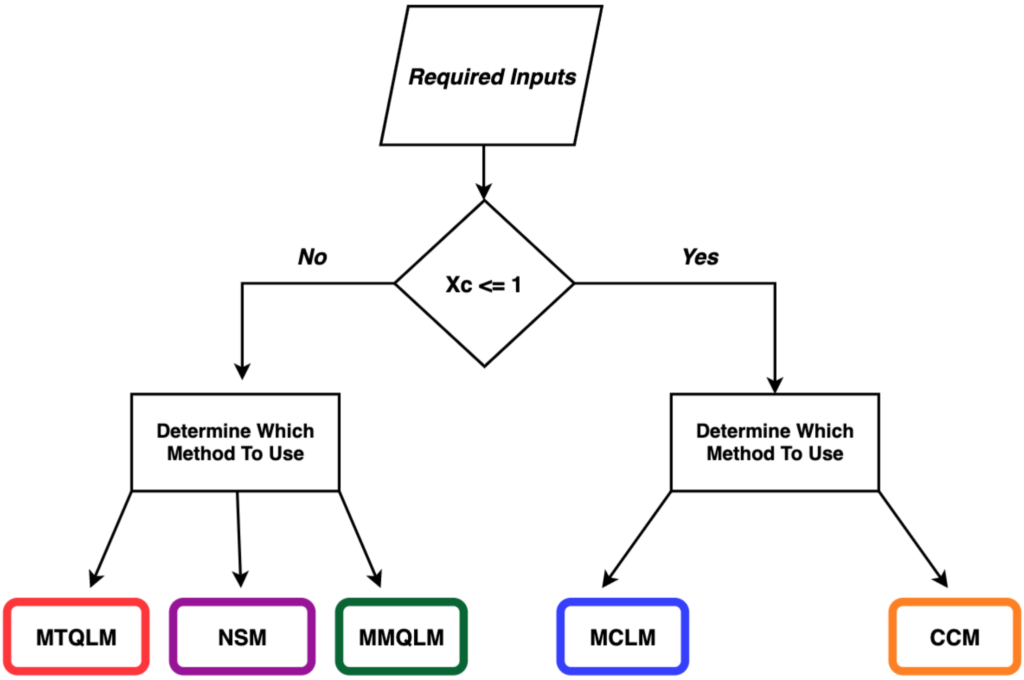


Figure 5.2: Superficial Flow Chart of the Workflow

5.3.1 A Sample Intersection Designed to Explain Linear Programming Approaches

For explaining linear programming approaches and methods developed in the study, a sample 4-legged intersection was designed with 4-phase and some assumptions (Table 5.1) were made. In figure 5.3, a sample intersection can be seen.

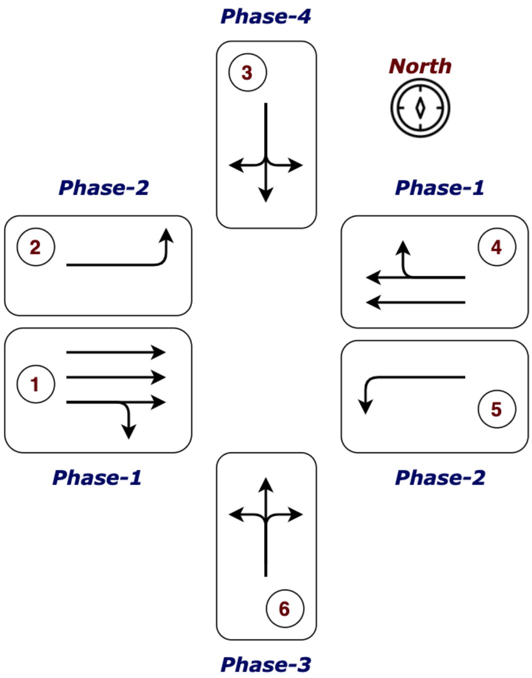


Figure 5.3: A Sample Intersection For Explaining Linear Programming Approaches Developed In the Study.

Table 5.1: Assumptions and Informations Of the Sample Intersection

Assumptions	
Initial Cycle Length	**135 seconds**
Minimum Effective Green Time Per Phase	**9 seconds**
Lost Time Per Phase	**2 seconds**
Yellow Time	**3 Seconds**
All Red Time	**1 Second**
Number Of Phases	**4**
Number Of Lane Groups	**6**
Pre-assumed Range Value (δ) For NSM	**5**

In order to understand the linear programming approaches developed, 4 different traffic scenarios were created by changing the traffic demands (volumes) of the directions at the intersection. These scenarios can be seen in the Table 5.2, 5.3, 5.4, 5.5, respectively.

Table 5.2: Scenario 1 for the sample intersection

	Left	**Through**	**Right**
West	300 (veh/hour)	1800 (veh/hour)	144 (veh/hour)
North	50 (veh / hour)	350 (veh/hour)	50 (veh/hour)
East	156 (veh/hour)	550 (veh/hour)	100 (veh/hour)
South	75 (veh/hour)	400 (veh/hour)	75 (veh/hour)

Table 5.3: Scenario 2 for the sample intersection

	Left	**Through**	**Right**
West	300 (veh/hour)	1667 (veh/hour)	133 (veh/hour)
North	60 (veh / hour)	320 (veh/hour)	60 (veh/hour)
East	156 (veh/hour)	550 (veh/hour)	100 (veh/hour)
South	50 (veh/hour)	250 (veh/hour)	50 (veh/hour)

Table 5.4: Scenario 3 for the sample intersection

	Left	**Through**	**Right**
West	180 (veh/hour)	720 (veh/hour)	164 (veh/hour)
North	30 (veh / hour)	120 (veh/hour)	30 (veh/hour)
East	72 (veh/hour)	250 (veh/hour)	50 (veh/hour)
South	44 (veh/hour)	200 (veh/hour)	44 (veh/hour)

Table 5.5: Scenario 4 for the sample intersection

	Left	**Through**	**Right**
West	300 (veh/hour)	1150 (veh/hour)	146 (veh/hour)
North	40 (veh / hour)	160 (veh/hour)	40 (veh/hour)
East	120 (veh/hour)	400 (veh/hour)	100 (veh/hour)
South	72 (veh/hour)	260 (veh/hour)	52 (veh/hour)

5.3.2 Nomenclatures For Linear Programming Approaches

P: Set of signal phases indexed by *p*

N: Number of phases

CG: Set of critical lane groups

G^p: Set of all allowable lane groups in the *p*th phase

n_i : Number of lanes in the lane group *i*

ω_i : Demand ratio of a lane group *i*

Ω : Total Demand Ratio of an intersection

R : Correction Coefficient

a_i : Allocation ratio of a lane group *i*

λ_i : Arrival rate per second for lane group *i*

θ_i : Lane-Based saturation low rate per second (veh/sec/lane) for lane group *i*

C: Total cycle length

L: Total lost time per cycle

x^p : Effective green time for the phase *p* (Decision Variable)

g^p_{min} : Minimum effective green time for the phase *p*

g^p_{max} : Maximum effective green time for the phase *p*

5.3.3 MTQLM - Minimize Total Queue Length Method

MTQLM is a linear programming approach which aims to minimize total remaining (residual) queue length after each cycle. Minimizing total remaining queue length also means that maximizing the total vehicles which can pass through the intersection (maximizing throughput). In other words, minimizing total remaining queue length and maximizing throughput objectives can be considered as the same objective. This algorithm is used only in the situation where ***Xc*** ratio is greater than 1 and aims to assign best effective green times for each phase to minimize total remaining queue length.

As mentioned before, if ***Xc*** is greater than 1, some critical lane groups will have residual queue after each cycle. This also means that effective green time is too short to discharge their queue at the end of the effective green period. As a result, residual queue continues to grow during the analysis period and permanent oversaturation occurs in some lane groups.

During the effective green period, as long as the queue length in the lane group remains greater than 0, the vehicles cross the intersection with the saturation flow rate. However, if there are no vehicles left in the queue, the vehicles pass through the intersection at their arrival rate.

In this method, while **MTQLM** try to discharge all the vehicles waiting in the queue for some lane groups, it also leaves some residual queue for other lane groups. This situation can also be considered from the following point of view. Since **MTQLM** tries to minimize the total remaining queue length, as soon as it clears the queue in any critical lane group, it will try to assign the appropriate effective time to other lane groups as well. The following inference can be made from this behaviour. While the product of the total saturation flow rate with the effective green time will be less than the total cumulative arriving vehicles for some critical lane groups, it will be equal to total cumulative arriving vehicles for other critical lane groups at most. This inference will be in **MTQLM** as an important constraint.

For every critical lane group $\boldsymbol{i} \in \boldsymbol{CG}$ and the corresponding phase $\boldsymbol{p} \in \boldsymbol{P}$, this constraint can be formulized in the following way,

$$\boxed{(Total\ Discharging)\ \mathbf{x^p}\theta_\mathbf{i}\mathbf{n_i} <= \lambda_\mathbf{i}\mathbf{C}\ \ (Cumulative\ Arrivings)} \tag{5.1}$$

Another constraint of this method is the minimum effective green times for phases. A traffic signal setting should consider pedestrian safety so that pedestrians can pass through the intersection safely [68]. Although pedestrians at intersections were not taken into account in the study, minimum effective green times for intersections were still determined.

For every phase ***p*** $\in$ ***P***, minimum effective time constraints can be formulized in the following way;

$$\boxed{\mathbf{x^p} >= \mathbf{g^p_{min}}} \tag{5.2}$$

The final constraint of this approach is that the total assignable effective green times equals the summation of the effective green times of each phase. The total assignable effective green times is also equal to the difference between cycle length and total lost time. This equality can be seen in the equation 5.3.

$$\boxed{\sum_{p\in P} \mathbf{x^p} = \mathbf{C} - \mathbf{L}} \tag{5.3}$$

Finally, The objective function and all contraints of **MTQLM** can be written as a whole in the following formal way;

$$\boxed{\begin{array}{ll} \text{minimize} \sum_{\mathbf{p}\in\mathbf{P}} \sum_{\mathbf{i}\in\mathbf{G^p}} (\lambda_\mathbf{i}\,\mathbf{C} - \mathbf{n_i}\,\mathbf{x^p}\,\theta_\mathbf{i}) & (Minimize\,Total\,Res.\,Queue) \\ \quad\quad\quad\quad \text{subject to} & \\ x^p\theta_i n_i \leq \lambda_i C\ \ ,\ \ \forall i \in CG & (Discharge\,Constraint) \\ x^p \geq g^p_{min}\ \ ,\ \ \forall p \in P & (Min.\,Green\,Constraint) \\ \sum_{p\in P} x^p = C - L\ \ ,\ \ \forall p \in P & (Cycle\,Constraint) \end{array}} \tag{5.4}$$

If Scenario 1 and Scenario 2 are used for the sample intersection was used for traffic signal optimization, critical lane groups can be found as 1, 2, 3 and 6. Lane group 4 and 5 are the non-critical lane groups in these scenarios. Thus, MTQLM optimizes the effective green times of each phase by using those critical lane groups and its constraints. With the purpose of numerical example, linear formulization of MTQLM for scenario 1 can be examined in appendices (Table A.1).

Effective green times optimized by MTQLM for each scenario can be found in table 5.6.

Table 5.6: The Effective Green Times optimized by MTQLM

Scenario	**Phase-1**	**Phase-2**	**Phase-3**	**Phase-4**	**Cycle Length (C)**
1	***48***	***22***	***20***	***33***	***135***
2	***45***	***22***	***23***	***33***	***135***

With these effective green times optimized by MTQLM, residue queue analysis can be made for Scenario 1 and Scenario 2. According to deterministic queueing, after **30 cycles**, total remaining queue (residual queue) lengths will be **364.5** and **56.1** for Scenario 1 and Scenario 2, respectively. Residual queue and its percentage results for each lane groups can be seen in table 5.7 and table 5.8, respectively.

Table 5.7: Residue Queue Results for Scenario 1 optimized by MTQLM

Lane Groups	***Total Arriving (veh)***	***Total Residue Queue (veh)***	**Residue Queue Percentage(%)**
1	**2187**	**27**	**1**
2	**337.5**	**7.5**	**2**
3	**506.25**	**11.4**	**2**
4	**731.25**	**0**	**0**
5	**175.5**	**0**	**0**
6	**618.75**	**318.6**	**51**

Table 5.8: Residue Queue Results for Scenario 2 optimized by MTQLM

Lane Groups	*Total Arriving (veh)*	*Total Residue Queue (veh)*	**Residue Queue Percentage(%)**
1	**2025**	**0**	**0**
2	**337.5**	**7.5**	**2**
3	**495**	**0**	**0**
4	**731.25**	**0**	**0**
5	**175.5**	**0**	**0**
6	**393.75**	**48.6**	**12**

HCM 2000 delay analysis can also be made for Scenario 1 and 2. If the analysis period is taken as 15 minutes, the average control delay of the intersection is **134.30** for Scenario 1 and **76.30** for Scenario 2. Average control delay for each lane group can be also be seen in figure 5.4 and figure 5.5.

Table 5.9: 15-Min HCM Delay for each lane group for Scenario 1

Lane Groups	*Average Control Delay (sec/veh)*	*Degree Of Saturation (v/c ratio)*
1	**67.17**	**1.01**
2	**115.0**	**1.02**
3	**99.8**	**1.02**
4	**35.65**	**0.51**
5	**58.53**	**0.53**
6	**548.39**	**2.06**

Table 5.10: 15-Min HCM Delay for each lane group for Scenario 2

Lane Groups	*Average Control Delay (sec/veh)*	*Degree Of Saturation (v/c ratio)*
1	**66.21**	**1.0**
2	**115.0**	**1.02**
3	**93.91**	**1.0**
4	**38.37**	**0.54**
5	**58.53**	**0.53**
6	**151.24**	**1.14**

If we look at the residual queue results for each lane group for Scenario 1 and Scenario

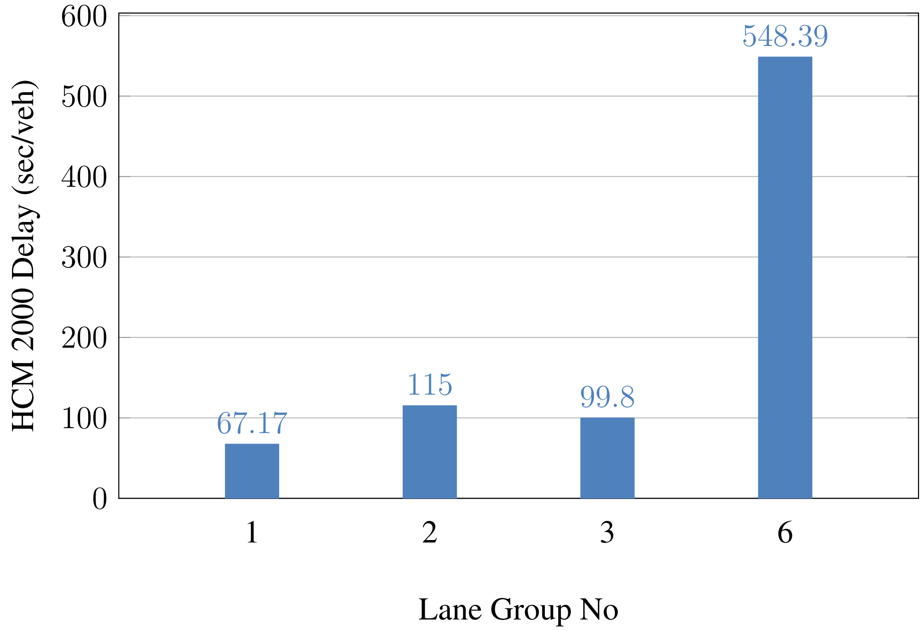


Figure 5.4: 15-Minute HCM Delay For Each Critical Lane Group for Scenario 1

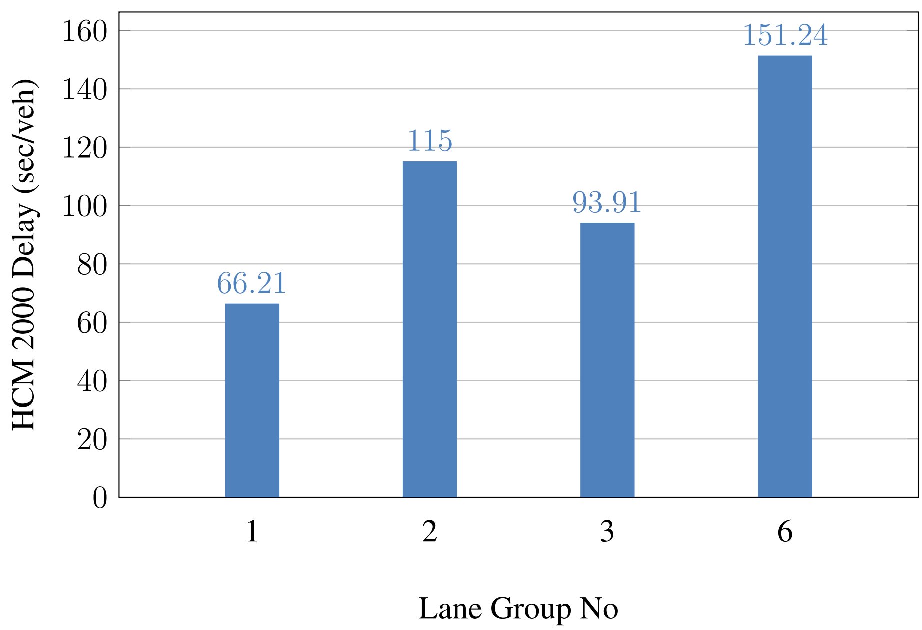


Figure 5.5: 15-Minute HCM Delay For Each Critical Lane Group for Scenario 2

2, MTQLM left unreasonable residual queues in some lane groups. For Scenario 1, almost all the total remaining queues remained at **lane group 6**.

In this case, 50% of the vehicles arriving to the lane group 6 will remain in the queue and as the analysis time increases, the vehicles will have to wait for one or more green periods to pass through the intersection. This can be understood from the fact that the degree of saturation of lane group 6 is a very high, 2.06 (Table 5.9).

According to the HCM 2000 delay calculation, permanent oversaturation causes high average control delay of the lane groups. Therefore, in the lane group 6, there were much higher delay (**548.39 sec/veh**) in the analysis time of 15 minutes compared to the other lane groups.

The same situation occurred for the **scenario 2**. MTQLM, for the scenario 1, left 12% residual queue in lane group 6, while in other lane groups it left almost no residual queues (Table 5.8). Because of this residual queue, a higher average control delay (**151.24 (sec/veh)**) occurred in lane group 6 than in the others, as in scenario 1.

Although MTQLM is a good algorithm for optimizing total residual queue by taking into account, the linear constraints that were determined, it has some disadvantages considering the residual queue in some lane groups. According to deterministic queueing model, while this algorithm leave zero queue in some lane groups and leave unreasonable residual queue in one or more lane groups, especially in traffic scenarios with high traffic demand. This situation results in permanent oversaturation in some lane groups as the duration of analysis period grows and therefore leads much higher average control delay in these lane groups compare to other lane groups with zero residual queue after each cycle. Moreover, if saturation flow rates are much lower for those lane groups with residual queue, more residual queue left for them and this situation becomes even worse. For the reasons mentioned here, we can say that the MTQLM algorithm penalizes some lane groups while minimizing the total remaining queue and does not behave fairly when assigning effective green times to the phases.

This algorithm was proposed for showing the deficiency of the linear programming model proposed by Liu [32]. In some traffic scenarios, MTQLM algorithm may result in higher average delay of the intersection and unbalanced delay in the critical lane groups. Therefore, new algorithm was proposed to assign effective green times in the purpose of behaving more fairly to the lane groups than MTQLM algorithm.

5.3.4 MMQLM - Minimize - Maximum Queue Length Method

MMQLM is another linear programming approach generated to overcome the unfairness problem in terms of residual queue and average control delay mentioned in the MTQLM section. MMQLM uses all constraints for assigning effective green times to the phases, just like MTQLM. The key differences between MMQLM and MTQLM are the brand new objective function.

While MTQLM aims to minimize total residual queue length at the intersection, MMQLM aims to minimize the maximum remaining queue length left in any lane group after each cycle. In other words, it tries to allocate the remaining total queue length to each critical lane group after each cycle. While the fair allocation of the remaining queue length to each critical lane group, this method uses the demand and allocation ratio of each critical lane group. The demand ratio, named by us for this method, is slightly different from flow ratio. As mentioned before, the flow ratio of a critical lane group is calculated by dividing the total traffic volume by the total saturation flow rate. Total saturation flow rate can be calculated by summing saturation flow rate of each lane in a lane group. In this thesis study, saturation flow rate per lane was assumed to be **1800 veh/hour/lane**. Therefore, total saturation flow rate is also equal to the multiplication of the number of lanes in a critical lane group and saturation flow rate (1800 veh/hour/lane) that was assumed. The demand ratio of a critical lane group , on the other hand, can be calculated by dividing the total traffic volume by the saturation flow rate per lane. The **demand ratio** of a critical lane group or non-critical lane group can be calculated by the following formula;

$$\boxed{\omega_\mathbf{i} = \mathbf{q_i}/\mathbf{s_i}} \tag{5.5}$$

MMQLM first calculates the demand ratios of each critical lane group and finds total demand ratio by sum of these individual demand ratios. To find the total demand ratio of an intersection, the following formula can be used;

$$\boxed{\sum_{\mathbf{i}\in\mathbf{CG}} \omega_{\mathbf{i}} = \mathbf{\Omega}} \tag{5.6}$$

Next, MMQLM calculates the allocation ratios of each critical lane group, dividing each individual demand ratio by total demand ratio of the intersection. The allocation ratio of a critical lane group can be calculated by the following formula;

$$\boxed{\mathbf{a_i} = \omega_{\mathbf{i}}/\mathbf{\Omega}} \tag{5.7}$$

Unlike the **MTQLM** approach, **MMQLM** tries to minimize the maximum residual queue that can occur in any critical lane group by placing each allocation ratio previously calculated in the denominator section of the residual queue formula of each critical lane group. In this way, if saturation flow rate per lane is same for all lanes, each lane in a critical lane group will have residual queue after each cycle of approximately same percentage. This is the major difference between **MMQLM** and **MTQLM**.

The linear programming formulization can be seen in the following formula;

$$\boxed{\begin{array}{cc} \text{minimize } \boldsymbol{max_{i\,\in\,CG}}((\lambda_i\, C - n_i\, x^p\, \theta_i)/a_i) & (Min.\, Max\, Res.\, Queue) \\ \text{subject to} & \\ x^p\theta_i n_i \leq \lambda_i C \ \ , \ \ \forall i \in CG & (Discharge\, Constraint) \\ x^p \geq g^p_{min} \ \ , \ \ \forall p \in P & (Min.\, Green\, Constraint) \\ \sum_{p\in P} x^p = C - L \ \ , \ \ \forall p \in P & (Cycle\, Constraint) \end{array}} \tag{5.8}$$

If Scenario 1 and Scenario 2 are used for the sample intersection was used for traffic signal optimization, critical lane groups can be found as 1, 2, 3, 6. Lane group 4 and 5 are the non-critical lane groups in these scenarios. Thus, MMQLM optimizes the effective green times of each phase by using those critical lane groups and its constraints. With the purpose of numerical example, linear formulization of MMQLM for scenario 1 can be examined in appendices (Table A.2).

Table 5.11: The Effective Green Times optimized by MMQLM

Scenario	Phase-1	Phase-2	Phase-3	Phase-4	Cycle Length (C)
1	*41*	*19*	*35*	*28*	*135*
2	*43*	*22*	*26*	*32*	*135*

Table 5.12: Residue Queue Results for Scenario 1 optimized by MMQLM

Lane Groups	*Total Arriving (veh)*	*Total Residue Queue(veh)*	*Residue Queue Percentage(%)*
1	**2187**	**342**	**15**
2	**337.5**	**52.5**	**15**
3	**506.25**	**86.4**	**17**
4	**731.25**	**0**	**0**
5	**175.50**	**0**	**0**
6	**618.75**	**93.6**	**15**

Effective green times optimized by MMQLM for each scenario can be found in table 5.11.

With these effective green times optimized by **MMQLM**, residual queue analysis can be made for Scenario 1 and Scenario 2. According to deterministic queueing, after **30 cycles**, total remaining queue (residual queue) lengths will be **574.5** and **116.1** for Scenario 1 and Scenario 2, respectively. Residual queue and its percentage results for each lane group can be seen in table 5.12 and table 5.13, respectively.

HCM 2000 delay analysis can also be made for Scenario 1 and 2. If the analysis period is taken as 15 minutes, the average control delay of the intersection is **127.09**

Table 5.13: Residue Queue Results for Scenario 2 optimized by MMQLM

Lane Groups	*Total Arriving (veh)*	*Total Residue Queue(veh)*	*Residue Queue Percentage(%)*
1	**2025**	**90**	**4**
2	**337.5**	**7.5**	**2**
3	**495**	**15**	**3**
4	**731.25**	**0**	**0**
5	**175.50**	**0**	**0**
6	**393.75**	**3.6**	**0.4**

Table 5.14: 15-Min HCM Delay for each lane group for Scenario 1 - MMQLM

Lane Groups	***Average Control Delay (sec/veh)***	***Degree of Saturation (v/c ratio)***
1	**136.93**	**1.19**
2	**173.64**	**1.18**
3	**168.63**	**1.21**
4	**42.32**	**0.59**
5	**65.29**	**0.62**
6	**150.68**	**1.18**

for Scenario 1 and **80.61** for Scenario 2. Average control delay for each lane group can be also be seen in figure 5.6 and figure 5.7.

Table 5.15: 15-Min HCM Delay for each lane group for Scenario 2 - MMQLM

Lane Groups	***Average Control Delay (sec/veh)***	***Degree of Saturation (v/c ratio)***
1	**81.01**	**1.05**
2	**115.0**	**1.02**
3	**103.33**	**1.03**
4	**40.29**	**0.57**
5	**58.53**	**0.53**
6	**105.28**	**1.01**

If we look at the residual queue results for each lane group for Scenario 1, using MMQLM the total residual queue after **30 cycles** was 574.5 and increased by 57.6% compared to the MTQLM. However, MMQLM left residual queues in all critical lane groups with almost same percentage. Because of these residual queue results, we can say that the MMQLM approach treats critical lane groups more fairly than the MTQLM approach. In addition, for scenario 1, the MMQLM approach reduced the average vehicle delay of the intersection by 5.4% compared to the MTQLM approach, to **127.09 sec/veh**. As a result, it can be said that in scenario 1, choosing the MMQLM approach is more appropriate in terms of the average vehicle delay of the intersection and fair allocation of residual queues.

Finally, for Scenario 2, **after 30 cycles**, using MMQLM, the total residual queue was

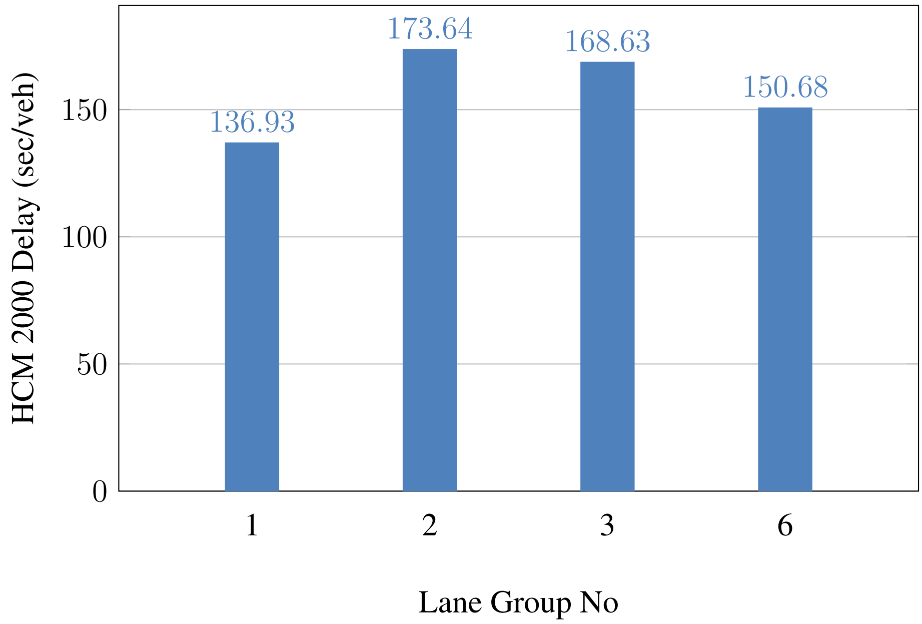


Figure 5.6: 15-Minute HCM Delay For Each Critical Lane Group for Scenario 1 - MMQLM

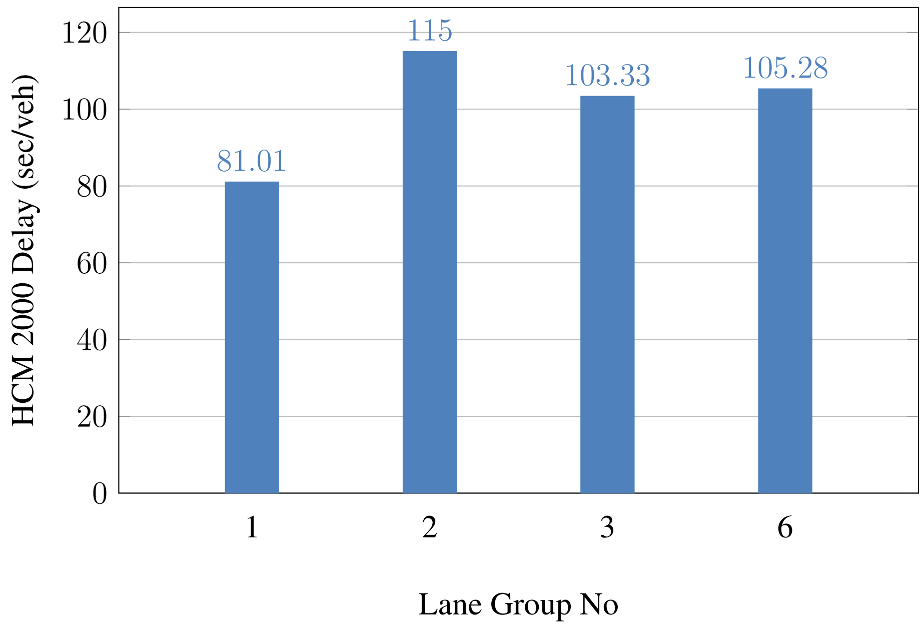


Figure 5.7: 15-Minute HCM Delay For Each Critical Lane Group for Scenario 1 - MMQLM

116.1 and increased by 106.9% compared to the MTQLM. However, unlike scenario 1, the average delay of the intersection in this scenario also increased by 5.6% compared to the MTQLM approach to 80.61. Therefore, it can be said that for scenario 2, it would be more appropriate to choose the **MTQLM approach**.

The following conclusion can be drawn from the comparisons here; As the unfairness of allocation residual queues in lane groups increases, it makes more sense to use the MMQLM approach than MTQLM. However, in some scenarios, using MTQLM may be more appropriate than MMQLM. Therefore, in any traffic scenario, it can be suggested to use and compare both algorithms at the same time. As a result of this comparison, the algorithm that reduces the average vehicle delay can be used.

Finally, from the comparisons made, it can be concluded that minimizing the total residual queue of the intersection does not always reduce the average control delay of the intersection in some traffic scenarios.

5.3.5 NSM - NeighbourHood Search Method

NeighbourHood Search Method , abbreviated as **NSM**, is not a linear programming approach. This approach is used in the second stage of optimization flow. In some traffic scenarios, using MTQLM approach results in lower average control delay than using MMQLM approach. However, in some traffic scenarios, using MMQLM approach gives much better results in terms of HCM 2000 delay compares to MTQLM. Therefore, given a traffic scenario, these two methods are always ran and compared in terms of average control delay. Whichever of these methods has found effective green times giving a lower average control delay, these effective green times are obtained first. This is the first stage of the workflow when **Xc** value is greater than 1.

In the second stage of the workflow, NSM is used to search for more appropriate effective green times in terms of delay. While searching more appropriate effective green times, NSM uses the obtained effective green times obtained in the first stage and tries to find to the best effective green times giving much better delay results than previous times within a pre-assumed range $[-\delta, +\delta]$. The workflow of the **NSM** is as follows;

1. Obtain the effective green times from **the first stage**
2. Determine the minimum and maximum borders of each effective green times according to the range value (δ)
3. According to the minimum and the maximum border of the phases determined, search the more appropriate green times for reducing average control delay in terms of HCM 2000 delay formula

While searching for new candidate effective green times, **NSM** creates nested for loops for the number of phases and tries all combinations according to the HCM 2000 delay formula at given ranges. Once the for loops are complete, it obtains the new effective green times for the phases. However, there are a few points to be emphasized about **NSM**. First, NSM may not always find more suitable effective green times than previous times found by MTQLM or MMQLM. Second, NSM can not guarantee that, by its very nature, it will always find a global optimum average control delay within a determined range. In addition, a simple **Exhaustive Search (Brute-Search)** algorithm can also be used for global minimum delay by trying all possible combinations with nested loops by taking account the minimum effective times without using these methods mentioned. However, as the number of phases and the total cycle length increases for an intersection, an exhaustive search algorithm may take a very long time to complete. In this study, a simple exhaustive search algorithm is also used for searching global optimum by creating nested for loops. According to the run time measurements made, it was decided that it would be appropriate to use the **NSM** algorithm to find the more appropriate effective green times.

Effective green times and methods used for scenario 1 and scenario 2 that find the average delay of the intersection lower can be seen in the table 5.16.

Table 5.16: Effective Green Times obtained from the first stage for scenario 1 and 2

Scenario	***Phase-1***	***Phase-2***	***Phase-3***	***Phase-4***	***Method***	***Average Control Delay (sec/veh)***
1	**41**	**19**	**35**	**28**	***MMQLM***	**127.09**
2	**45**	**22**	**23**	**33**	***MTQLM***	**76.30**

For scenario 1 and 2, NSM has optimized the effective green times by searching around the effective green times obtained in the first stage according to the previously determined range value δ (5 seconds). The newly found effective green times and average control delay can be seen from table 5.17.

Table 5.17: Effective Green Times obtained from NSM for scenario 1 and 2

Scenario	***Phase-1***	***Phase-2***	***Phase-3***	***Phase-4***	***Average Control Delay (sec/veh)***
1	**46**	**18**	**33**	**26**	**110.74**
2	**48**	**21**	**24**	**30**	**72.72**

Delay improvements percentage can also be seen in table 5.18.

Table 5.18: Delay Improvements using NSM in comparison to the first stage

Scenario	***NSM HCM Delay (sec/veh)***	***First Stage HCM Delay (sec/veh)***	***Delay Improvement (%)***
1	**110.74**	**127.09**	**12.86%**
2	**72.72**	**76.30**	**4.69%**

An **exhaustive search algorithm** has also been created for these scenarios to show that NSM does not always find the global minimum average control delay of the intersection. In addition, the comparison of the runtime of each of these algorithms was also made. The number of iterations of the algorithms can be expressed with the following equations to better understand the run time measurements.

The number of the iterations of NSM can be calculated by equation 5.9.

$$\boxed{\text{(NSM)} \quad \text{number of iterations} = (2\delta)^N} \tag{5.9}$$

The number of the iterations of exhaustive search algorithm can be calculated by equation 5.10.

$$\boxed{\text{(Exhaustive Search)} \quad \text{number of iterations} = (g^p_{max})^N} \tag{5.10}$$

For this intersection, g^p_{min} was determined as **9 seconds**. Therefore, the maximum effective green time of a phase can take, g^p_{max}, occurs when the other phases have the minimum effective green time. g^p_{max} can be calculated by equation 5.11;

$$\boxed{g^p_{max} = C - L - (N-1) * g^p_{min}} \tag{5.11}$$

If we look closely to g^p_{max} equation, g^p_{max} increases as the cycle length of the intersection increases. Therefore, as the cycle length of the intersection and the number of phases increases, the number of iterations of exhaustive search algorithm also increases and thus it takes more and more time to complete its run. On the other hand, the number of iterations of NSM slightly increase and depend on only the number of phases. As a conclusion from this, it can be said that **the exhaustive search algorithm** will take much longer than **NSM**.

In this study, each algorithm, **NSM and exhaustive search algorithm**, was run **10 times** to compare the run times between them, and the average of the 10 different run times found was taken into account. For this intersection where the number of phases (N) is **4**, the minimum effective green time is **9 seconds** and the range value δ is **5 seconds**, the run time measurements and the number of iterations of these algorithms can be seen in table 5.19.

Table 5.19: Run time measurements and the number of iterations of NSM and Exhaustive Search

NSM Mean Run Time (sec)	***Exhaustive Search Mean Run Time(sec)***	***NSM no.of iterations***	***Exhaustive Search no.of iterations***
0.13	**35.34**	**10.000**	**84.934.656**

As can be understood from the measurements in the table, the NSM algorithm completed its run approximately **270 times** faster than the exhaustive search algorithm to improve average control delay for this intersection. If we look at the number of iterations in table 5.19, it is obvious that the run time difference between these algorithms will be higher as the cycle length (C) and the number of phases (N) increase. As a result, in this study, because we aim to develop fast and effective algorithms for traffic signal optimization, it is a very reasonable choice to use **NSM** according to these measurements.

As mentioned earlier, as long as the cycle length is constant, the NSM algorithm does not guarantee that it will always find the global minimum delay. To illustrate this fact, NSM and exhaustive search algorithm were run for Scenario 1 and Scenario 2 and average control delays of the intersection are shown in table 5.20.

Table 5.20: Comparison between NSM and Exhaustive Search in terms of average control delay

Scenario	*NSM* *Average Control Delay (sec/veh)*	*Exhaustive Search* *Average Control Delay (sec/veh)*	*Is Global Minimum Found ?*
1	**110.74**	**107.53**	**NO**
2	**72.72**	**72.72**	**YES**

When the **Xc** ratio is less than or equal to 1, two different linear programming approaches were generated in the study. In the following two sections, MCLM and CCM approaches will be explained.

5.3.6 MCLM - Minimize Cycle Length Method

In some traffic scenarios, especially in traffic scenarios with low traffic demand, initial cycle length may cause the vehicles on the critical lane groups to wait more time for the next green period. This situation may also cause higher average control delay of each critical lane group and the intersection. Therefore, **MCLM** was developed to measure the effect of a dynamic cycle length, which is less than the specified initial cycle length on the average control delay.

MCLM is a linear programming approach to aim finding minimum total cycle length, leaving no residual queue in any critical lane group. The objective function of this method is **minimizing total cycle length**.

When the cycle length is long at an intersection, the effective green times will also be long for any critical lane group. This situation also means the multiplication of the total saturation flow rate with the effective green time of a phase is greater than total cumulative vehicles in a cycle. For this reason, MCLM has the constraint for each critical lane group that the total number of cumulative arriving vehicles in a cycle can be at most equal to the product of the total saturation flow rate and the effective green time. This constraint will lead the MCLM method to find the minimum cycle length, leaving no residual queue in any critical lane group according to deterministic queueing model. This constraint can also be thought of as the opposite of the logic behind the discharge constraint used in the **MTQLM** and **MMQLM** approaches described in the previous sections. This constraint can be formulized in the following way;

$$\boxed{(Cumulative\ Arrivings)\lambda_{\mathbf{i}}\mathbf{C}\ <=\ \mathbf{x^{p}}\theta_{\mathbf{i}}\mathbf{n_{i}}(Total\ Discharging)} \tag{5.12}$$

Minimum effective green times constraint are also included in this method. Thanks to this constraint, when the traffic demand of each critical lane group is much more low, this algorithm ensures that the minimum effective green times will be set for each phase. Equation 5.13 formulates linear programming constraints and objective function of **MCLM**.

$$\boxed{\begin{array}{cc} \text{minimize} \sum_{p\in P} x^{p} & (Minimize\, Total\, Cycle\, Length) \\ \text{subject to} & \\ \lambda_{\mathbf{i}}\mathbf{C} <= \mathbf{x^{p}}\theta_{\mathbf{i}}\mathbf{n_{i}}\ \ ,\ \forall lg \in CG & (Discharge\, Constraint) \\ x^{p} \geq g^{p}_{min}\ \ ,\ \forall p \in P & (Min\, Green\, Constraint) \end{array}} \tag{5.13}$$

If Scenario 3 and Scenario 4 are used for the sample intersection was used for traffic signal optimization, critical lane groups can be found as 1, 2, 3 and 6. Lane group 4 and 5 are the non-critical lane groups in these scenarios. Thus, MCLM optimizes the effective green times of each phase by using those critical lane groups and its constraints. With the purpose of numerical example, linear formulization of MCLM for scenario 3 can be examined in appendices (Table A.3).

Effective green times optimized by MCLM for each scenario can be found in table 5.21.

Table 5.21: Effective Green Times Found By MCL Method for Scenario 3 and 4

Scenario	*Phase-1*	*Phase-2*	*Phase-3*	*Phase-4*	*Cycle Length (C)*
3	***9***	***9***	***9***	**9**	***48***
4	***14***	***10***	***13***	**9**	***58***

Recall that, according to the deterministic queuing model, MCLM leaves no queues in any lane group. For this reason, only the delay analysis for MCLM will be done in this section. In the previous section, the average control delay of the intersection was calculated with only **15 minutes of analysis period**. However, in this section and in the next section, the delay calculation will also be made with **1 hour of analysis period** and the effect of duration of analysis period on average control delay will be discussed.

With the effective green times found by MCLM, average control delay of the intersection for Scenario 3 was calculated **27.49 (sec/veh)** when the duration of analysis period is 15 minutes and **28.70 (sec/veh)** when the analysis period is 1 hour. Likewise, for Scenario 4, average control delay is **44.80 (sec/veh)** when the duration of analysis period is 15 minutes, **61.14 (sec/veh)** when the analysis period is 1 hour (Figure 5.8).

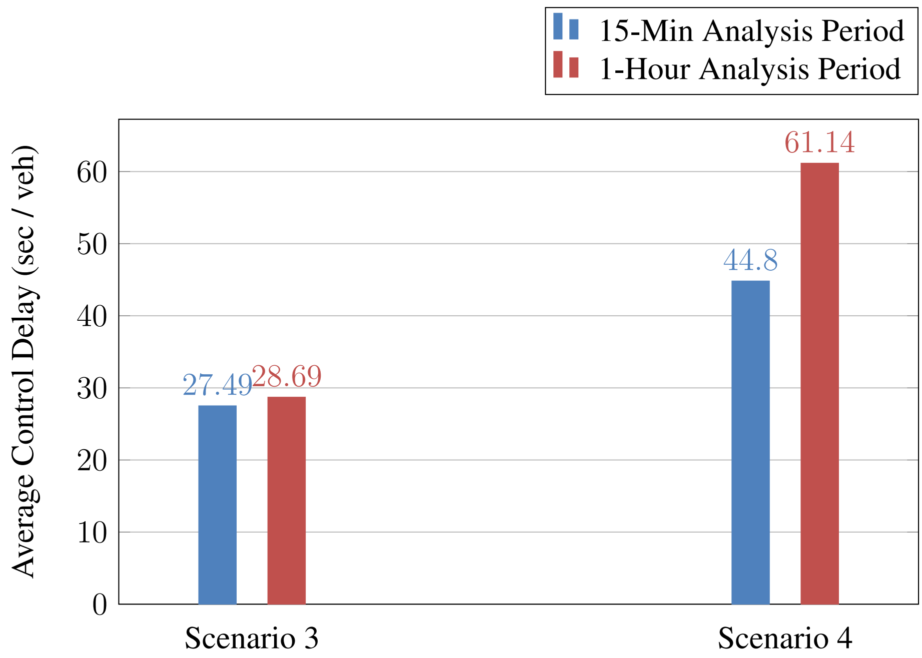


Figure 5.8: 15-Min and 1 Hour Average Control Delay of the intersection for Scenario 3 and Scenario 4 - MCLM

When the graph is examined carefully, it is seen that for Scenario 3, the average control delay of the intersection is almost the same when the analysis period is 15-Min and 1-hour. On the other hand, for Scenario 4, with the analysis period of 15 minutes, the average control delay is **44.8**, with the analysis period of 1-hour it is **61.14** with **36.4%** increase. To understand the reason behind these results, we need to consider each scenario and examine the average vehicle delays that occur in each lane group of the scenario carefully.

The 15-min and 1-hour average control delays of each lane group in the Scenario 3 can be seen in the tables below.

Table 5.22: 15-Min and 1-Hour Delay for Scenario 3 - MCLM

Lane Groups	*15-Min Delay(sec/veh)*	*1-Hour Delay(sec/veh)*	*Degree of Saturation (v/c ratio)*
1	***27.95***	***28.83***	***0.85***
2	***23.53***	***23.66***	***0.53***
3	***23.53***	***23.66***	***0.53***
4	***19.40***	***19.41***	***0.44***
5	***17.94***	***17.95***	***0.21***
6	***41.87***	***46.91***	***0.85***

In table 5.22, if we look at the degree of saturation of each lane group, it can be seen that the degree of saturation of lane group 2, 3, 4 and 5 are **quite low**. Therefore, when the duration of analysis period is 15-Min and 1-Hour, the average control delays in these lane groups are mostly due to **uniform delays (d1)** and the effect of **incremental delay** is quite low. As a result, there is only a slight difference between the delays in the analysis period of 15-minute and 1-hour in these lane groups. On the other hand, the degree of saturation of lane group 1 and 6 is 0.85 and is considerably higher than other lane groups. For this reason, a certain difference was observed between the delays in the 15-minute analysis period and the 1-hour analysis period in these lane groups. This difference is due to the **incremental delay (d2)** of HCM that occurs in addition to the **uniform delay (d1)** of HCM. However, the effect of incremental delay is still lower, as the degree of saturation is not very close to 1.

The following conclusion can be drawn from the analysis here; For scenario 3, the average control delay at the intersection is due to significantly uniform delays, and therefore there is little delay difference between the analysis period of 15 minutes and the analysis period of 1-hour.

The 15-min and 1-hour average control delays of each lane group in the Scenario 4 can be seen in table 5.23.

Table 5.23: 15-Min and 1-Hour Delay for Scenario 4 - MCLM

Lane Groups	*15-Min Delay(sec/veh)*	*1-Hour Delay(sec/veh)*	*Degree of Saturation (v/c ratio)*
1	***45.56***	***66.8***	***0.99***
2	***67.12***	***98.68***	***0.97***
3	***51.33***	***58.5***	***0.86***
4	***22.15***	***22.18***	***0.58***
5	***24.89***	***24.93***	***0.39***
6	***56.38***	***76.41***	***0.95***

In this scenario, the degree of saturation of lane group 4 and 5 (non-critical lane groups) is quite lower than other lane groups and the average control delays is due to mostly uniform delays, as in scenario 3. Therefore, for these lane groups, there is almost no difference in delays between the analysis period of 15-Minutes and the analysis period of 1-Hour. On the other hand, the degree of saturation of lane group 1, 2, 3, 6 is very close to 1 and thus for these lane groups, there are considerably big differences in delays between the analysis period of 15 Minutes and 1 Hour. The reason behind these increases in delays is that when the degree of saturation is very close to 1, in addition to **uniform delay**, the effect of **incremental delay** is also quite high. For this reason, if the degree of saturation of a lane group is close to 1, as the duration of analysis period increases, the average control delay of that lane group will continue to grow.

As a result, in Scenario 4, these increases in the delays of the lane groups are also the reason for the **36% increase** in the average control delay of the intersection between the analysis period of 15 minutes and the analysis period of 1 hour.

According to the analysis made here, the major disadvantage of MCLM is that it causes high degrees of saturation and in turn high effect of incremental delays in many traffic scenarios, especially in critical lane groups, as a result of its objective function.

The logic behind MCLM algorithm is quite similar to the model proposed by Srisurin

[31]. Srisurin, only used deterministic queuing logic while evaluating his model. However, minimizing cycle length objective may be misleading in terms of average vehicle delay at the intersection. To show the deficiency of Srisurin's model, for under-saturated conditions, a new approach, CCM, has been proposed besides MCLM. In the next section, this approach will be discussed.

5.3.7 CCM - Correction Coefficient Method

CCM is a linear programming approach developed for under-saturated condition (Xc <= 1), inspired by the MTQLM approach. In some traffic scenarios, the Xc ratio drops below 1 (under-saturated condition). In such cases, the **MTQLM** approach can not create a feasible solution (infeasible solution). This is because of the discharge constraint and cycle constraint included in the MTQLM approach. The right-hand side of discharge constraints and cycle constraint prevents the method from finding a feasible solution. In under-saturated conditions, while MTQLM tries to satisfy all discharge constraints, it can not satisfy the cycle constraint. For example, if an intersection has a pre-determined assignable cycle length of **123 seconds**, and due to discharge constraints, if the sum of the green times (decision variables) can be a maximum of **112 seconds**, then MTQLM will not find a feasible solution at all (Figure 5.9). Therefore, in under-saturated condition, CCM was developed with the idea of multiplying the right side of the inequality by a correction coefficient in order to use the logic behind the **MTQLM**. In other words, **CCM** is the adaptation of the **MTQLM** approach for the under-saturated conditions.

```
Discharge_Constraint___1: 1.5 Phase_1 <= 67.5

Discharge_Constraint___2: 0.5 Phase_2 <= 13.5

Discharge_Constraint___3: 0.5 Phase_3 <= 6.75

Discharge_Constraint___4: 0.5 Phase_4 <= 13.5

Cycle_Constraint: Phase_1 + Phase_2 + Phase_3 + Phase_4 = 123
```

Figure 5.9: MTQLM - Infeasible Solution Situation

Unlike **MCLM**, CCM approach keep the cycle length constant and share the appropriate effective green times to each phase by adding a simple coefficient, **correction coefficient**, to the right-hand side of the inequality of the discharge constraint of the **MTQLM** approach. The purpose of the development of this method is to compare the effect of cycle length kept constant and the dynamic cycle length on HCM 2000 average control delay in under-saturated conditions.

Correction coefficient (R) for this method is strongly related with the **Xc** ratio and total available cycle length (C - L). In the case where the correction coefficient is not added to the constraint, the maximum value that the sum of the effective green time of each phase can take will be smaller than the total assignable effective green time. For this reason, the correction coefficient is added to the constraint to prevent this situation.

Correction coefficient found for this method can be calculated by the following formula;

$$\boxed{\mathbf{R = 1/Xc}} \tag{5.14}$$

With all these constraints and objective function, linear formulization of the CCM can be seen in equation 5.15;

$$\boxed{\begin{array}{ll} \text{minimize} \sum_{\mathbf{p \in P}} \sum_{\mathbf{i \in G^p}} \lambda_\mathbf{i}\, \mathbf{C} - \mathbf{n_i}\, \mathbf{x^p}\, \theta_\mathbf{i} & (Minimize\, Total\, Res.\, Queue) \\ \quad\quad\quad \text{subject to} & \\ x^p \theta_i n_i \leq R \lambda_i C \ , \ \forall i \in CG & (Discharge\, Constraint) \\ x^p \geq g^p_{min} \ , \ \forall p \in P & (Min.\, Green\, Constraint) \\ \sum_{p \in P} x^p = C - L \ , \ \forall p \in P & (Cycle\, Constraint) \end{array}} \tag{5.15}$$

With the purpose of numerical example, linear formulization of CCM for scenario 3 can be examined in appendices (Table A.4).

In table 5.24, it can be seen the effective green times of each phase optimized by CCM, for scenario 3 and scenario 4.

Table 5.24: Effective green times found by CCM for Scenario 3 and 4

Scenario	*Phase-1*	*Phase-2*	*Phase-3*	*Phase-4*	*Cycle Length (C)*
3	***38***	***24***	***37***	**24**	***135***
4	***39***	***28***	***34***	**22**	***135***

With the effective green times found by CCM, average control delay of the intersection for Scenario 3 was calculated **46.04 (sec/veh)** when the duration of analysis period is 15 minutes and **46.10 (sec/veh)** when the analysis period is 1 hour. Likewise, for Scenario 4, average control delay is **54.63 (sec/veh)** when the duration of analysis period is 15 minutes, **55.70 (sec/veh)** when the analysis period is 1 hour (Figure 5.10).

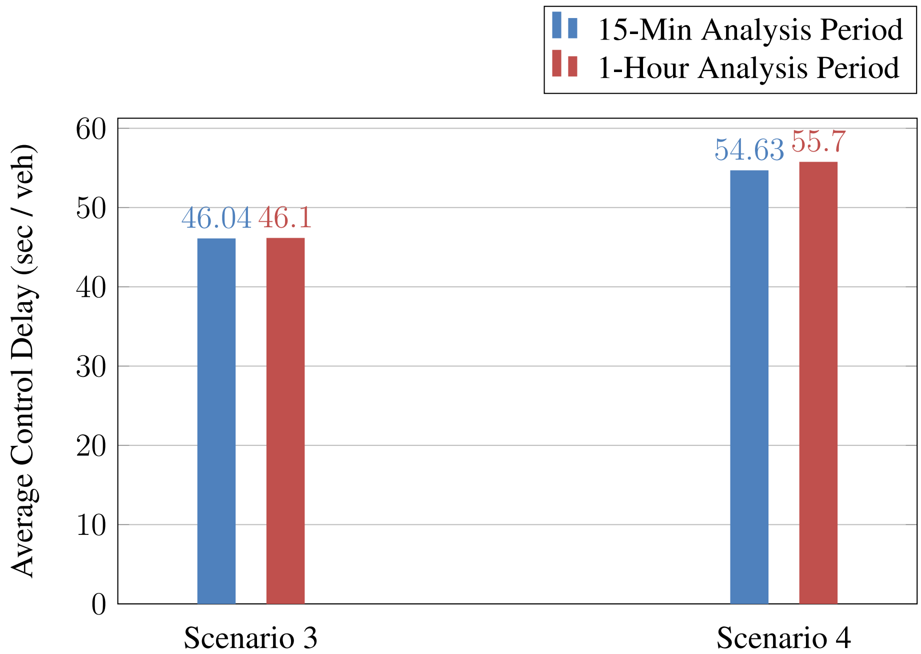


Figure 5.10: 15-Min and 1-Hour average control delay of the intersection for Scenario 3 and Scenario 4 - CCM

If the average control delays at the intersection for Scenario 3 and Scenario 4 are examined carefully, it can be seen that the increase between the delays after the 15-minute analysis period and the delays after the 1-hour period analysis is **quite low**. To investigate the reason for this, the degree of saturation of each lane group and the

Table 5.25: 15-Min and 1-Hour Delay for Scenario 3 - CCM Method

Lane Groups	***15-Min Delay(sec/veh)***	***1-Hour Delay(sec/veh)***	***Degree of Saturation (v/c ratio)***
1	***43.03***	***43.04***	***0.57***
2	***57.69***	***57.87***	***0.56***
3	***57.69***	***57.87***	***0.56***
4	***38.76***	***38.76***	***0.3***
5	***49.16***	***49.17***	***0.23***
6	***47.33***	***47.43***	***0.58***

Table 5.26: 15-Min and 1-Hour Delay for Scenario 4 - CCM Method

Lane Groups	***15-Min Delay(sec/veh)***	***1-Hour Delay(sec/veh)***	***Degree of Saturation (v/c ratio)***
1	***50.21***	***50.48***	***0.83***
2	***67.49***	***69.62***	***0.80***
3	***76.37***	***80.18***	***0.82***
4	***41.23***	***41.24***	***0.48***
5	***47.7***	***47.71***	***0.32***
6	***65.55***	***68.5***	***0.85***

delay values after the analysis period of 15 minutes and the delay values after the analysis period of 1 hour can be examined. For scenario 3 and 4, the average control delays of each lane group can be seen in table 5.25 and table 5.26, respectively.

If we examine the average control delays of each lane group for Scenario 3 and Scenario 4, degrees of saturation of none of those lane groups are very close to 1 and thus the difference between 15-Min delays and 1-Hour delays is quite low. As a result, this situation leads to only a slight difference between 15-Min average control delay of the intersection and 1-Hour average control delay of the intersection.

5.3.8 Delay Comparison Between MCLM and CCM

In previous two sections, two methods, **MCLM and CCM**, were introduced for under-saturated conditions and delays occurred in the lane groups were examined. In this section, we will compare MCLM and CCM in terms of delay for Scenario 3 and Scenario 4.

For Scenario 3 and 4, using MCLM and CCM, the average control delays of the intersection occurred after 15-Min and 1-Hour and differences between those delays can be seen in figure 5.11 and figure 5.12, respectively.

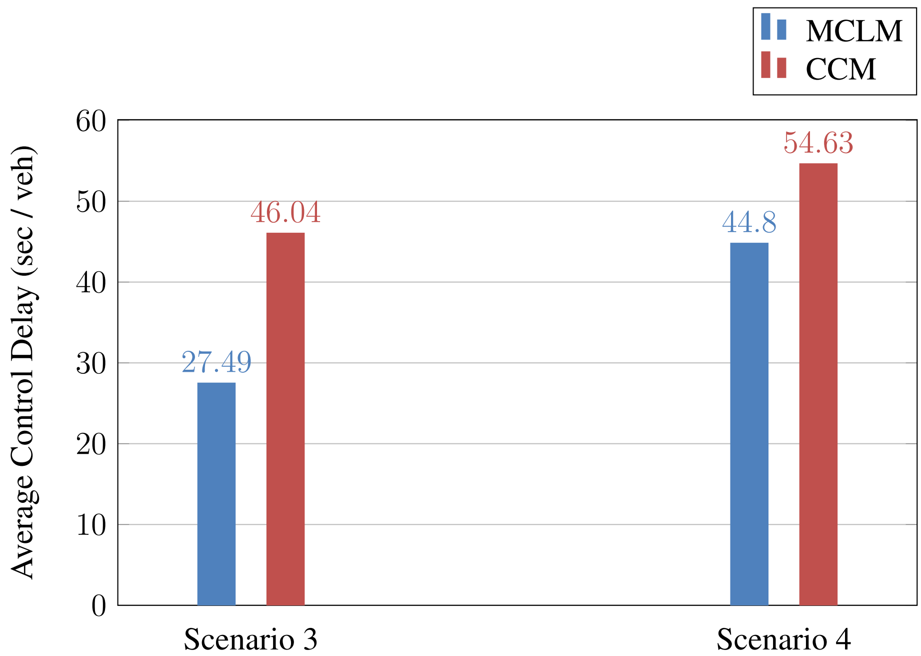


Figure 5.11: 15-Min Average Control Delay of the intersection for Scenario 3 and Scenario 4 - CCM and MCLM

If 15-Min average control delays of the intersection are examined carefully, it can be seen that MCLM gave much lower delay results than CCM. MCLM reduced the 15-Min average control delay of the intersection compared to CCM by 40.3% and 18% for Scenario 3 and Scenario 4, respectively.

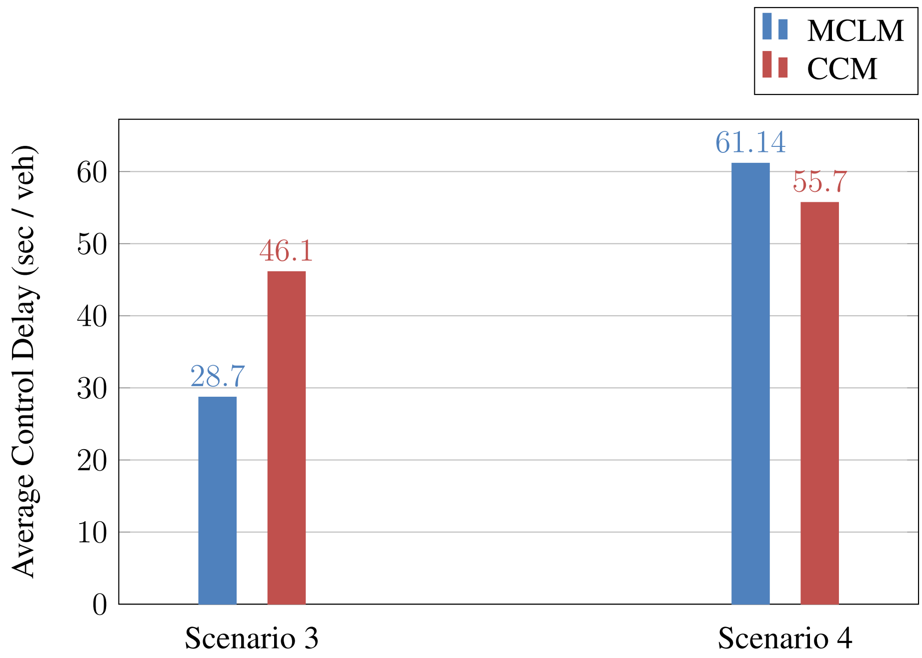


Figure 5.12: 1 Hour Average Control Delay of the intersection for Scenario 3 and Scenario 4 - CCM and MCLM

For scenario 3, MCLM reduced 1-Hour average control delay of the intersection by 37.7% compared to CCM. However, CCM gave much lower 1-hour delay results compared to MCLM for scenario 4. CCM reduced 1-Hour average control delay 8.9% for scenario 4. These results are very interesting and considerable because for Scenario 4, MCLM gives much better delay results than CCM when the analysis period 15-Min, while CCM gives much better when the analysis period 1-Hour. The reason behind these results was discussed in section which explains MCLM. The major disadvantage of **MCLM** is that it causes high degrees of saturation in critical lane groups while trying to reduce the total cycle length. For this reason, in most scenarios, CCM gives much better delay results compared to MCLM considering 1-Hour average control delay.

The following conclusion can be drawn from here; For under-saturated conditions, the 15-Min HCM delay is not sufficient to compare between these methods. For this reason, in the following sections of the thesis, 1-hour average control delays of the intersections will be taken into consideration when comparing CCM and MCLM.

Finally, the situations here are not valid for over-saturated conditions. For example, if MTQLM performs better on delay results of 15 minutes than MMQLM, it will also give better results in 1 hour delay results.

5.3.9 The Flow of Entire Algorithm

In the previous sections, all methods developed in the thesis study were introduced. In **experiments chapter**, we will test three types of intersection for evaluating these methods in detail. General flow of the entire algorithm can be seen in figure 5.13.

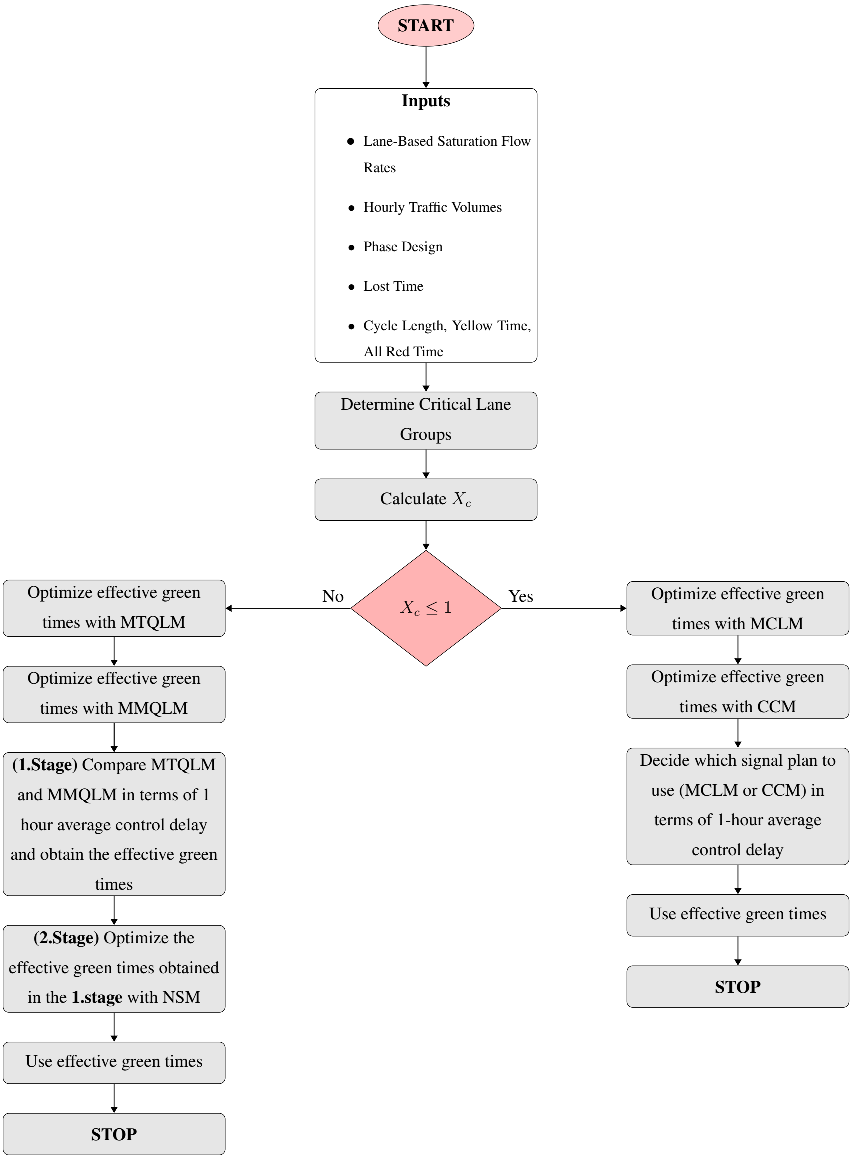


Figure 5.13: The Flow Diagram of Entire Algorithm

CHAPTER 6

EXPERIMENTS

In this section, analyzes results of 3 different types of intersections designed to compare the linear programming methods previously described within the scope of the thesis study are included.

- In the first subsection, analyses made for **4-legged** intersection, named **Intersection 1**, designed for comparing linear programming methods and delay results are discussed. In order to test under-saturated and over-saturated conditions and methods, **10 different traffic scenarios** were generated. In this subsection, average delay results of the intersection are included.
- In the second subsection, analyses made for **T-Type** intersection, named **Intersection 2** designed for comparing linear programming methods and delay results are discussed. For this intersection, **12 different traffic scenarios** were generated for comparing methods developed. In this subsection, average delay results of the second intersection are included.
- In the last subsection, another **4-legged intersection**, named Intersection 3, which differs from the geometry of the Intersection 1, has been designed and the most comprehensive analyzes have been carried out for this intersection. 12 different scenarios were created for this intersection. In order to measure the effect of different phase plans (design) on delays, 2 different phase plans were created and tested, unlike Intersection 1 and 2. At the end of this subsection, Intersection 3 and the optimized the effective green times obtained were tested after calibration in the **PTV VISSIM**, a microsimulation software, and the delay results obtained by the **HCM 2000** delay method were compared with the **VISSIM** delay results.

6.0.1 Experiment 1: Analyzing the Intersection 1

Intersection 1 is a type of 4-legged intersection. In this intersection, there are no repeated lane groups. In other words, each lane group gain right of way only once a phase. Also, there are some lanes that can be shared by multiple movements. For example, in the north direction, right, left and straight (through) movements share the same lane and create a lane group. There are 4 phases for this intersection. The phases where each approach gain right of way were designed by considering potential conflicts carefully. The intersection geometry can be seen in figure 6.1.

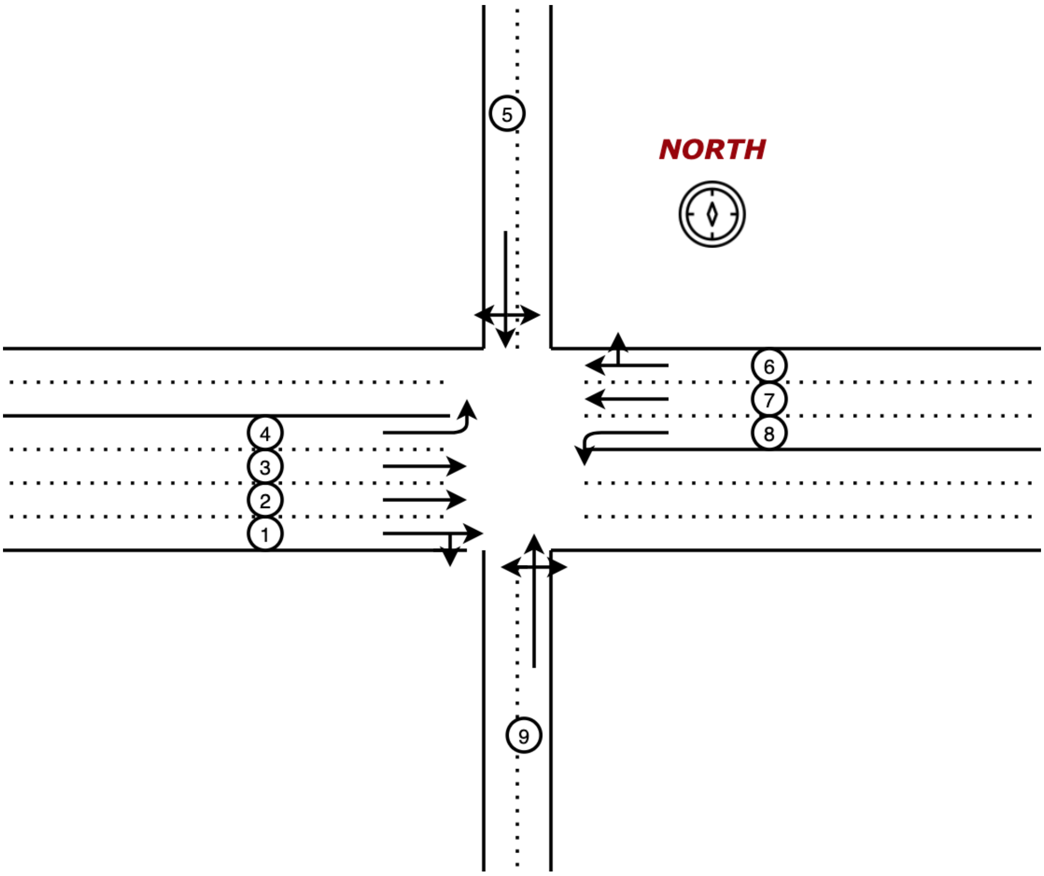


Figure 6.1: Intersection 1 - Geometry

As seen in the figure 6.1, Intersection 1 has a total of 9 lanes. According to the movements in lanes, these lanes combine to form 6 different lane groups. These lane groups and each corresponding phases for each lane group for Intersection 1 can be seen in figure 6.2.

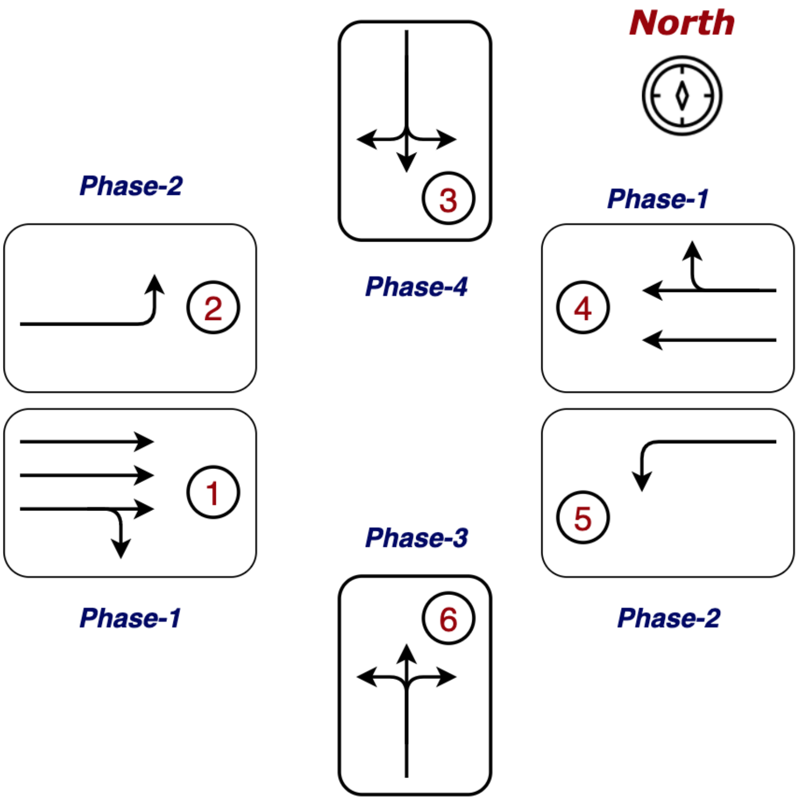


Figure 6.2: Intersection 1 - Phase Design and Lane Groups

- Lane 1,2 and 3 enable vehicles to make straight movements from west to east direction and right turns from west to south direction and they create lane group 1. The vehicles on Lane Group 1 gain right of way in Phase 1. **Lane Group 1** can also be named as **west-straight-right** or simply **w-s-r** because it enables vehicles make straight and right movements from west direction.

- Lane 4 enables vehicles make left turns from west to north direction and this lane create lane group 2. The vehicles on Lane Group 2 gain right of way in Phase 2. **Lane Group 2** can also be named as **west-left** or simply **w-l** because it enables vehicles make left turns from west direction.

- Lane 5 enables vehicles make left turns, right turns and straight (through) movements and this lane creates lane group 3. The vehicles on Lane Group 3 gain right of way in Phase 4. **Lane Group 3** can also be named as **north-straight-left-right** or simply **n-s-l-r** because it enables vehicles make left turns, right turns and through movements from north direction.

- Lane 6 and 7 enable vehicles make straight (through) movements from east to west direction and right turns from east to north direction and they create lane group 4. The vehicles on Lane Group 4 gain right of way in Phase 1. **Lane Group 4** can also be named as **east-straight-right** or simply **e-s-r** because it enables vehicles make straight and right movements from east direction.

- Lane 8 enables vehicles make left turns from east to south direction and this lane creates lane group 5. The vehicles on Lane Group 5 gain right of way in Phase 2. **Lane Group 5** can also be named as **east-left** or simply **e-l** because it enables vehicles make left turns from east direction.

- Lane 9 enables vehicles to make left turns, right turns and straight (through) movements and this lane creates lane group 6. The vehicles on Lane Group 6 gain right of way in Phase 3. **Lane Group 6** can also be named as **south-straight-left-right** or simply **s-s-l-r** because it enables vehicles make left turns, right turns and through movements from south direction.

For evaluating the linear programming approaches mentioned in chapter 5, 10 different traffic scenarios were generated for this intersection. First, a base scenario was determined, including hourly traffic volumes from each direction and each movement. After that, other traffic scenarios were generated from this base scenario by increasing or decreasing traffic volumes of base scenario with some percentage. Hourly traffic volumes of the base scenario by **direction** can be seen in table 6.1.

As explained before, lane groups enable vehicles to make left turns, right turns and straight movements. For example, lane group 1 has straight movements and right turns from the west direction. Therefore, total hourly traffic volume of this lane group can be found by summing the hourly traffic volume of straight movements and right turns. Hourly traffic volumes of the base scenario by **lane group** can be seen in table 6.2.

Hourly traffic volumes of each lane group in each scenario can be seen in table 6.3.

Table 6.1: Hourly Traffic Volumes(veh/hour) By Direction - Base Scenario - Intersection 1

	Directions		
	Left	***Straight***	***Right***
West	*300 (veh/hour)*	*1260 (veh/hour)*	*180 (veh/hour)*
North	*50 (veh/hour)*	*200 (veh/hour)*	*50 (veh/hour)*
East	*120 (veh/hour)*	*450 (veh/hour)*	*50 (veh/hour)*
South	*60 (veh/hour)*	*360 (veh/hour)*	*60 (veh/hour)*

Table 6.2: Hourly Traffic Volumes (veh/hour) By Lane Group - Base Scenario - Intersection 1

Lane Group No	**Volume (veh/hour)**
1 (w-s-r)	*1440*
2 (w-l)	*300*
3 (n-s-l-r)	*300*
4 (e-s-r)	*500*
5 (e-l)	*120*
6 (s-s-l-r)	*480*

Table 6.3: Hourly Traffic Volumes (veh/hour) By Lane Group For Each Scenario - Intersection 1

	Lane Groups					
Scenario No	***1 (w-s-r)***	**2 (w-l)**	**3 (n-s-l-r)**	**4 (e-s-r)**	**5 (e-l)**	**6 (s-s-l-r)**
1	*864*	*180*	*180*	*300*	*72*	*288*
2	*1008*	*180*	*210*	*350*	*84*	*336*
3	*1152*	*300*	*300*	*500*	*120*	*384*
4	*1296*	*300*	*240*	*500*	*120*	*384*
5 (Base Scenario)	*1440*	*300*	*300*	*500*	*120*	*480*
6	*1440*	*330*	*330*	*500*	*120*	*480*
7	*1728*	*300*	*300*	*500*	*120*	*528*
8	*1728*	*360*	*360*	*600*	*144*	*576*
9	*1872*	*375*	*390*	*650*	*156*	*576*
10	*1944*	*375*	*450*	*650*	*156*	*624*

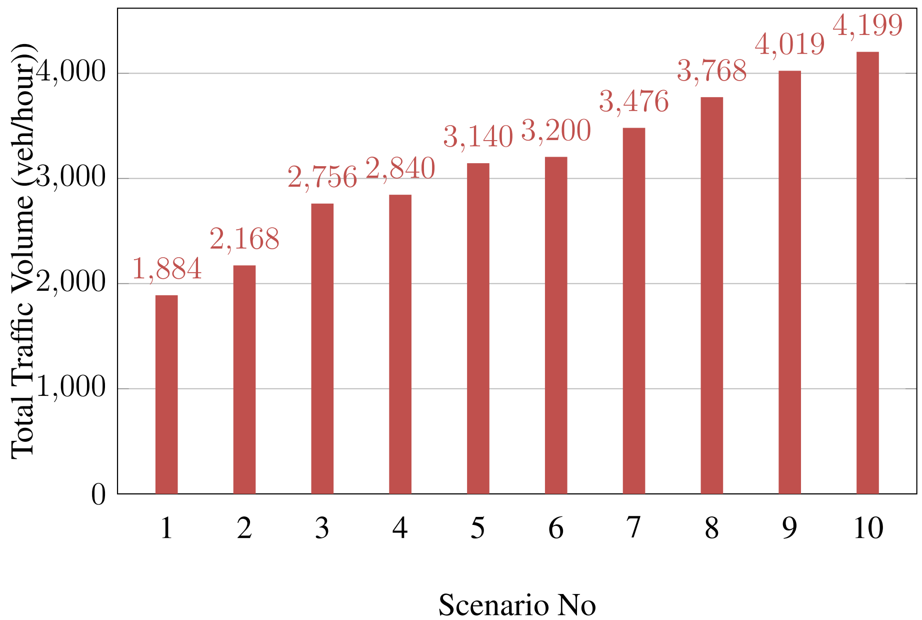


Figure 6.3: Total Traffic Volume of the Intersection 1 For Each Scenario

Finally, in table 6.4 , it can be seen the additional informations and assumptions for Intersection 1.

Table 6.4: Assumptions and Informations Of Intersection 1

Assumptions	
Initial Cycle Length	**135 seconds**
Minimum Effective Green Time Per Phase	**9 seconds**
Lost Time Per Phase	**2 seconds**
Yellow Time	**3 Seconds**
All Red Time	**1 Second**
Number Of Phases	**4**
Preassumed Range Value (δ) For NSM	**5**

6.0.1.1 Analyzing Under-Saturated Conditions

In this section, under-saturated conditions that occur according to the determined phase design and scenarios will be examined and a 1-hour delay comparison will be made using the previously mentioned MCLM and CCM. According to the determined hourly traffic volumes, under-saturated conditions were observed in Scenario 1 to Scenario 6 in the intersection. For this reason, MCLM and CCM are only run in these scenarios.

Optimized effective green times of each phase and **1-hour HCM 2000** average control delay of the intersection with the help of **MCLM** can be seen in table 6.5.

Table 6.5: Optimized Effective Green Times - MCLM - Intersection 1

Scenario No	*Phase-1*	*Phase-2*	*Phase-3*	*Phase-4*	*Cycle Length (C)*	*1-Hour Average Control Delay (sec/veh)*
1	***9***	***9***	***9***	***9***	***48***	***28.69***
2	***9***	***9***	***9***	***9***	***48***	***59.62***
3	***12***	***9***	***12***	***9***	***54***	***63.53***
4	***14***	***10***	***13***	***9***	***58***	***61.14***
5	***24***	***15***	***24***	***15***	***90***	***87.75***
6	***32***	***22***	***32***	***22***	***120***	***98.51***

Optimized effective green times of each phase and **1-hour HCM 2000** average control delay of the intersection with the help of **CCM** can be seen in table 6.6.

Table 6.6: Optimized Effective Green Times - CCM - Intersection 1

Scenario No	Phase-1	Phase-2	Phase-3	Phase-4	Cycle Length (C)	1-Hour Average Control Delay (sec/veh)
1	38	24	37	24	135	46.10
2	39	22	36	26	135	47.75
3	34	28	33	28	135	59.04
4	39	28	34	22	135	55.70
5	38	24	37	24	135	76.41
6	36	26	35	26	135	100.59

The comparative delay results of **CCM and MCLM** according to scenarios can be examined in the bar plot in figure 6.4.

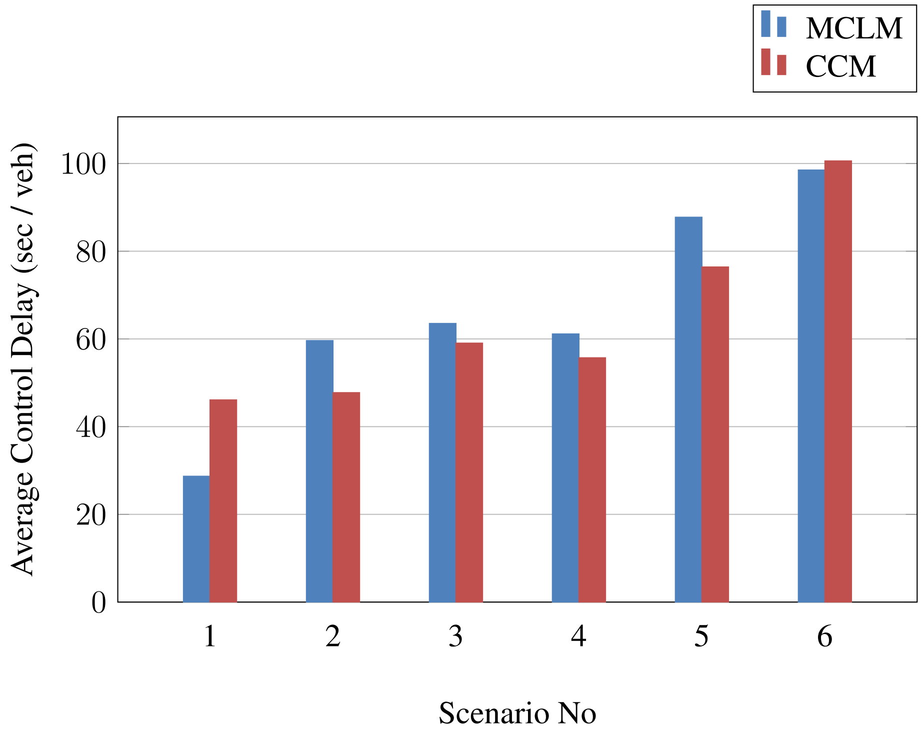


Figure 6.4: 1 Hour Average Control Delay Comparison - Undersaturated Conditions - MCLM and CCM

As seen in figure 6.4, MCLM gave less average control delay results in Scenario 1 and Scenario 6 whereas CCM gave much better delay results in Scenario 2, 3, 4 and 5. From the measurements here, it can be concluded that in these scenarios, using CCM for signal optimization is more appropriate than MCLM.

If we examine the delay measurements of each scenario separately;

- In Scenario 1, the average delay was calculated as **28.69 (sec/veh)** with effective green times optimized by MCLM, whereas this average delay was calculated as **46.10 (sec/veh)** with CCM. Based on this result, MCLM reduced the average delay by approximately **37.8%** compared with CCM. The reason behind the result here is that hourly traffic volumes in Scenario 1 are quite low. Also, degrees of saturation of each lane group are not very close to 1 with these effective green times optimized by MCLM (see chapter 5). Thus, the effect of incremental delay of HCM 2000 is relatively low compared to uniform delay of HCM 2000. Finally, for this scenario, CCM gave much higher delay than MCLM as it caused the vehicles to wait more time for next green period. The degrees of saturation and 1-hour delay for each lane group caused by both MCLM and CCM can be examined in table 6.7 and table 6.8.

- In Scenario 2, the average delay was calculated as **47.75 (sec/veh)** with effective green times optimized by CCM, whereas this average delay was calculated as **59.62 (sec/veh)** with MCLM. Based on this result, CCM reduced the average delay by approximately **19.9%** compared to MCLM. The reason behind this result is that MCLM caused high degrees of saturation (very close to 1) in some lane groups (critical lane groups) (see section). The degrees of saturation and 1-hour delay for each lane group caused by both MCLM and CCM can be examined in table 6.9 and table 6.10.

- In Scenario 3, the average delay was calculated as **59.04 (sec/veh)** with effective green times optimized by CCM, whereas this average delay was calculated as **63.53 (sec/veh)** with MCLM. Based on this result, CCM reduced the average delay by approximately **7.1%** compared to MCLM.

- In Scenario 4, the average delay was calculated as **55.70 (sec/veh)** with effective green times optimized by CCM, whereas this average delay was calculated as

61.14 (sec/veh) with MCLM. Based on this result, CCM reduced the average delay by approximately **8.9%** compared to MCLM.

- In Scenario 5, the average delay was calculated as **76.41 (sec/veh)** with effective green times optimized by CCM, whereas this average delay was calculated as **87.75 (sec/veh)** with MCLM. Based on this result, CCM reduced the average delay by approximately **12.9%** compared to MCLM.

- In Scenario 6, the average delay was calculated as **98.51 (sec/veh)** with effective green times optimized by MCLM, whereas this average delay was calculated as **100.59 (sec/veh)** with CCM. Based on this result, MCLM reduced the average delay by approximately **2.1%** compared to CCM. For this scenario, both MCLM and CCM caused high degrees of saturation of some lane groups because X_c is very close to 1. Therefore, they gave approximately same delay results for this scenario.

Table 6.7: 1-Hour Delay and Degrees of Saturation For Each Lane Group - Scenario 1 - MCLM

Lane Groups	***1-Hour Delay(sec/veh)***	***Degree of Saturation (v/c ratio)***
1	***28.83***	***0.85***
2	***23.66***	***0.53***
3	***23.66***	***0.53***
4	***19.41***	***0.44***
5	***17.95***	***0.21***
6	***46.91***	***0.85***

Table 6.8: 1-Hour Delay and Degrees of Saturation For Each Lane Group - Scenario 1 - CCM

Lane Groups	*1-Hour Delay(sec/veh)*	*Degree of Saturation (v/c ratio)*
1	*43.04*	*0.57*
2	*57.87*	*0.56*
3	*57.87*	*0.56*
4	*38.76*	*0.30*
5	*49.17*	*0.23*
6	*47.43*	*0.58*

Table 6.9: 1-Hour Delay and Degrees of Saturation For Each Lane Group - Scenario 2 - MCLM

Lane Groups	*1-Hour Delay(sec/veh)*	*Degree of Saturation (v/c ratio)*
1	*72.06*	*1.0*
2	*23.66*	*0.53*
3	*26.61*	*0.62*
4	*20.41*	*0.52*
5	*18.38*	*0.25*
6	*113.32*	*1.0*

Table 6.10: 1-Hour Delay and Degrees of Saturation For Each Lane Group - Scenario 2 - CCM

Lane Groups	*1-Hour Delay(sec/veh)*	*Degree of Saturation (v/c ratio)*
1	*44.07*	*0.65*
2	*62.16*	*0.61*
3	*57.71*	*0.61*
4	*38.69*	*0.34*
5	*52.07*	*0.29*
6	*53.24*	*0.7*

6.0.1.2 Analyzing Over-Saturated Conditions

In this section, over-saturated conditions that occur according to the determined phase design and scenarios will be examined and a 1-hour delay comparison will be made using the previously mentioned MTQLM, MMQLM and NSM. According to the determined hourly traffic volumes, over-saturated conditions were observed in Scenario 7 to Scenario 10 at the intersection. For this reason, MTQLM, MMQLM and NSM are only run in these scenarios.

Optimized effective green times of each phase and **1-hour HCM 2000** average control delay of the intersection with the help of **MTQLM** can be seen in table 6.11.

Table 6.11: Optimized Effective Green Times - MTQLM - Intersection 1

Scenario No	*Phase-1*	*Phase-2*	*Phase-3*	*Phase-4*	*Cycle Length (C)*	*1-Hour Average Control Delay (sec/veh)*
7	*43*	*22*	*36*	*22*	*135*	**126.06**
8	*43*	*27*	*26*	*27*	*135*	**272.02**
9	*46*	*28*	*20*	*29*	*135*	**395.24**
10	**48**	**28**	**14**	**33**	**135**	**720.97**

Optimized effective green times of each phase and **1-hour HCM 2000** average control delay of the intersection with the help of **MMQLM** can be seen in table 6.12.

Table 6.12: Optimized Effective Green Times - MMQLM - Intersection 1

Scenario No	*Phase-1*	*Phase-2*	*Phase-3*	*Phase-4*	*Cycle Length (C)*	*1-Hour Average Control Delay (sec/veh)*
7	*41*	*22*	*38*	*22*	*135*	**146.72**
8	*38*	*24*	*38*	*23*	*135*	**267.35**
9	*39*	*24*	*36*	*24*	*135*	**347.98**
10	**38**	**22**	**37**	**26**	**135**	**465.14**

Optimized effective green times of each phase and **1-hour HCM 2000** average control delay of the intersection with the help of **NSM** can be seen in table 6.13.

Table 6.13: Optimized Effective Green Times - NSM - Intersection 1

Scenario No	*Phase-1*	*Phase-2*	*Phase-3*	*Phase-4*	*Cycle Length (C)*	*1-Hour Average Control Delay (sec/veh)*
7	***44***	***21***	***37***	***21***	***135***	**120.30**
8	***43***	***22***	***36***	***22***	***135***	**212.21**
9	***44***	***22***	***34***	***23***	***135***	**279.80**
10	**43**	**21**	**34**	**25**	**135**	**390.07**

The comparative delay results of MTQLM, MMQLM and NSM according to scenarios can be examined in the bar plot in figure 6.5.

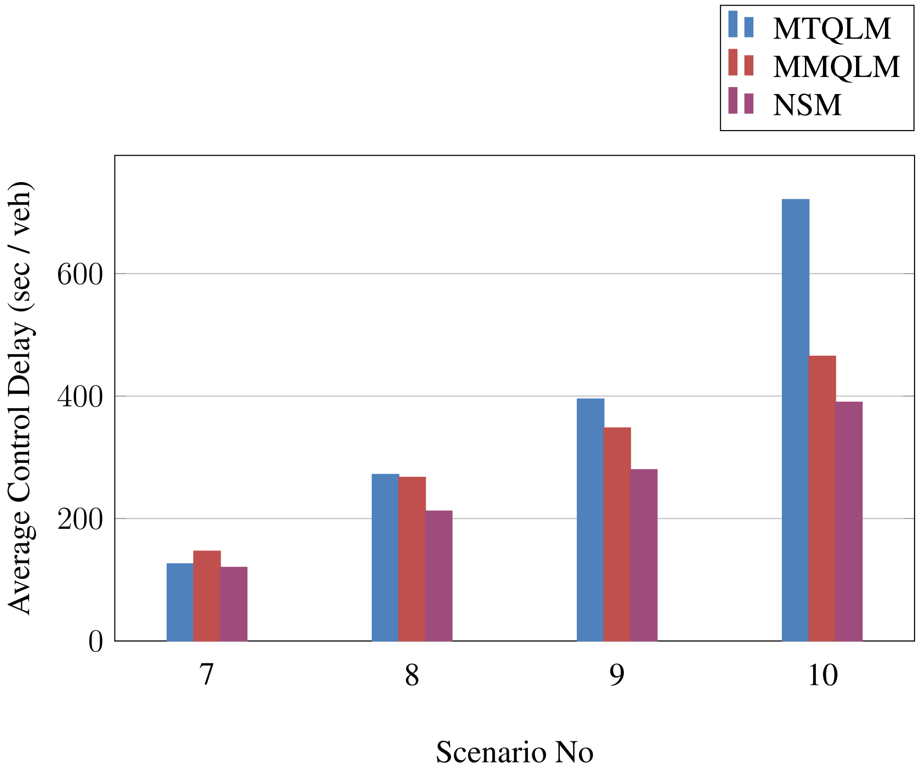


Figure 6.5: 1 Hour Average Control Delay Comparison - Oversaturated Conditions - MTQLM, MMQLM and NSM - Intersection 1

As seen in figure 6.5, as the hourly traffic volumes of each critical and non-critical lane groups increases, MQTLM gave worse delay results compared to other methods. As mentioned in chapter 5, this is because unfairness allocation of total residual queue. In addition, NSM gave much better delay results in every scenario compared to MTQLM, MMQLM.

If we examine the delay measurements of each scenario separately;

- In Scenario 7, the average delay was calculated as **126.06 (sec/veh)** with effective green times optimized by MTQLM, whereas this average delay was calculated as **146.72 (sec/veh)** with MMQLM. Based on this result, MTQLM reduced the average delay by approximately **14%** compared to MMQLM. So, optimized effective green times of MTQLM were selected in the first stage. In the second stage, NSM optimized effective green times obtained in the first stage within an interval, determined by predetermined range value (δ) and found new optimized effective green times. Finally, using NSM, the average delay was calculated as **120.30 (sec/veh)**. Based on this result, NSM reduced the average delay by approximately **4.6%** compared to MTQLM.
- In Scenario 8, the average delay was calculated as **267.35 (sec/veh)** with effective green times optimized by MMQLM, whereas this average delay was calculated as **272.02 (sec/veh)** with MTQLM. Based on this result, MMQLM reduced the average delay by approximately **1.7%** compared to MTQLM. Finally, the average delay was calculated as **212.21 (sec/veh)** with the more optimized effective green times of NSM. Based on this result, NSM reduced the average delay by approximately **20.6%** compared to MMQLM.
- In Scenario 9, the average delay was calculated as **347.98 (sec/veh)** with effective green times optimized by MMQLM, whereas this average delay was calculated as **395.24 (sec/veh)** with MTQLM. Based on this result, MMQLM reduced the average delay by approximately **12%** compared to MTQLM. Finally, the average delay was calculated as **279.80 (sec/veh)** with the more optimized effective green times of NSM. Based on this result, NSM reduced the average delay by approximately **19.6%** compared to MMQLM.
- In Scenario 10, the average delay was calculated as **465.14 (sec/veh)** with effective green times optimized by MMQLM, whereas this average delay was calculated as **720.97 (sec/veh)** with MTQLM. Based on this result, MMQLM reduced the average delay by approximately **35.5%** compared to MTQLM. Finally, the average delay was calculated as **390.07 (sec/veh)** with the more optimized effective green times of NSM. Based on this result, NSM reduced the

average delay by approximately **16.1%** compared to MMQLM.

Finally, delay comparison results of **exhaustive search algorithm** and **NSM** can be seen in table 6.14. According to the results in table 6.14, NSM could find the global minimum in Scenario 7 and 8, while it could not find the global minimum in Scenario 9 and 10.

Table 6.14: Delay Comparison Between Exhaustive Search and NSM - Intersection 1

Scenario No	*1-Hour Delay (sec/veh) NSM*	*1-Hour Delay (sec/veh) Exhaustive Search*	*Is Global Minimum Found ?*
7	*120.30*	*120.30*	**Yes**
8	*212.21*	*212.21*	**Yes**
9	*279.80*	*265.69*	**No**
10	*390.07*	*356.92*	**No**

Comparison of runtime measurements between **NSM and Exhaustive Search** are presented in chapter 5. Therefore, in this experiment and following experiments, comparison of runtime measurements between these methods will not be made.

6.0.1.3 Preferred Methods For Intersection 1

According to the delay comparisons made in the previous sections, the methods to be preferred for each scenario and effective green times can be seen in table 6.15.

Table 6.15: Preferred Methods - Intersection 1

Scenario No	*Phase-1*	*Phase-2*	*Phase-3*	*Phase-4*	*Cycle Length (C)*	*Preferred Method*	*1-Hour Delay (sec/veh)*
1	*9*	*9*	9	9	48	*MCLM*	28.69
2	*39*	*22*	36	26	135	*CCM*	47.75
3	*34*	*28*	33	28	135	*CCM*	59.04
4	*39*	*28*	34	22	135	*CCM*	55.70
5	38	24	37	24	135	*CCM*	76.41
6	32	22	32	22	120	*MCLM*	98.51
7	44	21	37	21	135	*NSM*	120.30
8	43	22	36	22	135	*NSM*	212.21
9	44	22	34	23	135	*NSM*	279.80
10	43	21	34	25	135	*NSM*	390.07

6.0.2 Experiment 2 - Analyzing Intersection 2

Intersection 2 has phase design that includes some repeated lane groups. For example, some lane groups gain right of way in more than one phase. Unlike the first intersection, all movement separated from each other and there are additional lanes for all movements. This intersection was designed as **T-shaped** intersection. The intersection geometry can be seen in figure 6.6.

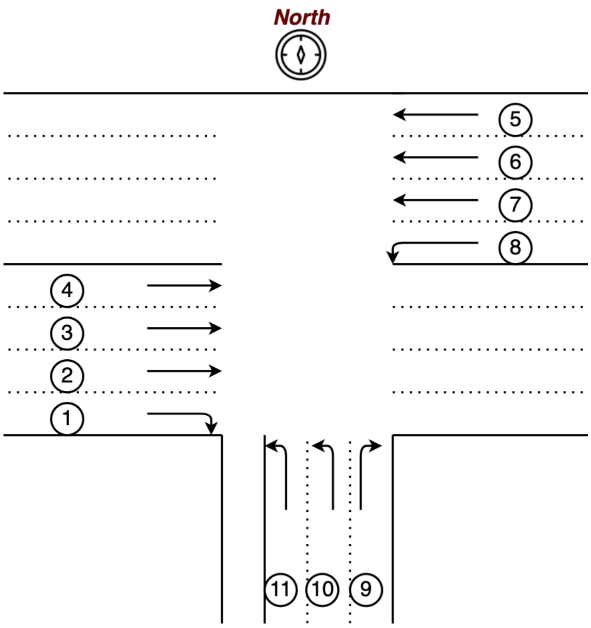


Figure 6.6: Intersection 2 - Geometry

As seen in figure 6.6, Intersection 2 has a **total of 11 lanes**. According to the movements, these lanes combine to form **6 different lane groups**. These lane groups determined for Intersection 2 can be seen in figure 6.7. The phase design for Intersection 2 can be seen in figure 6.8.

- Lane Group 1 includes only lane 1 and this lane group gains right of way in only one phase, **Phase-1**. This lane group is also named as **west-right** or simply **w-r**.

- Lane Group 2 includes lane 2, 3 and 4 and this lane group gains right of way in only one phase, **Phase-1**. This lane group is also named as **west-straight** or simply **w-s**.

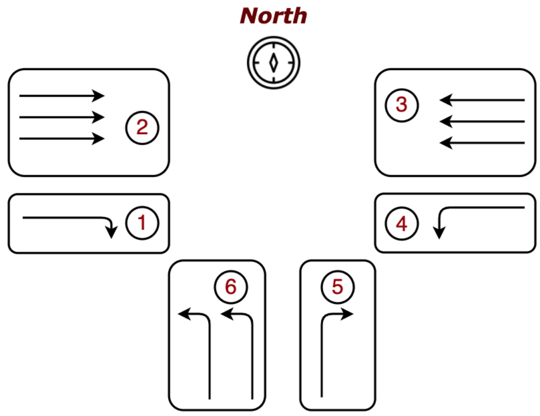


Figure 6.7: Intersection 2 - Lane Groups

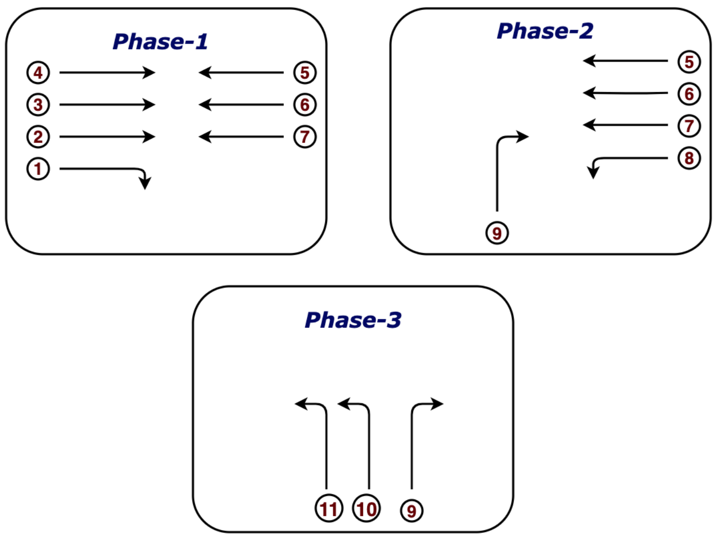


Figure 6.8: Intersection 2 - Phase Design

- Lane Group 3 includes lane 5,6 and 7 and this lane group gains right of way in two consecutive phases, **Phase-1** and **Phase-2**. This lane group is also named as **east-straight** or simply **e-s**.

- Lane Group 4 includes only lane 8 and this lane group gains right of way in only one phase, **Phase-2**. This lane group is also named as **east-left** or simply **e-l**.

- Lane Group 5 includes only lane 9 and this lane group gains right of way in two consecutive phases, **Phase-2** and **Phase-3**. This lane group is also named as **south-right** or simply **s-r**.

- Lane Group 6 includes lane 10 and 11 and this lane group gains right of way in only one phase, **Phase-3**. This lane group is also named as **south-left** or simply **s-l**.

For evaluating the linear programming approaches mentioned in chapter 5, **12 different traffic scenarios** were generated for this intersection. Strategy for generating traffic scenarios is exactly the same as for Intersection 1. Hourly traffic volumes of the base scenario by **direction** can be seen in table 6.16.

Table 6.16: Hourly Traffic Volumes(veh/hour) By Direction - Base Scenario - Intersection 2

	Directions		
	Left	*Straight*	*Right*
West	-	*1560 (veh/hour)*	*150 (veh/hour)*
North	-	-	-
East	*500 (veh/hour)*	*900 (veh/hour)*	-
South	*900 (veh/hour)*	-	*100 (veh/hour)*

Hourly traffic volumes of the base scenario by **lane group** can be seen in table 6.17.

Table 6.17: Hourly Traffic Volumes(veh/hour) By Lane Group - Initial Scenario - Intersection 2

Lane Group No	*Volume (veh/hour)*
1 (w-r)	*150*
2(w-s)	*1560*
3(e-s)	*900*
4(e-l)	*500*
5(s-r)	*100*
6(s-l)	*900*

Hourly traffic volumes of each lane group in each scenario can be seen in table 6.18.

Table 6.18: Hourly Traffic Volumes (veh/hour) By Lane Group For Each Scenario - Intersection 2

	Lane Groups					
Scenario No	**1 (w-r)**	**2 (w-s)**	**3 (e-s)**	**4 (e-l)**	**5 (s-r)**	**6 (s-l)**
1	*105*	*1092*	*630*	*350*	*70*	*630*
2	*135*	*1248*	*810*	*350*	*90*	*720*
3	*135*	*1404*	*810*	*450*	*90*	*810*
4	*150*	*1248*	*900*	*500*	*100*	*900*
5	*150*	*1560*	*900*	*500*	*100*	*900*
6	*150*	*1560*	*900*	*550*	*100*	*900*
7	*150*	*1716*	*900*	*500*	*100*	*900*
8	*150*	*1560*	*900*	*500*	*100*	*1080*
9	*150*	*1872*	*990*	*550*	*110*	*990*
10	*180*	*1872*	*1080*	*600*	*120*	*1080*
11	*165*	*2028*	*990*	*600*	*110*	*1170*
12	*165*	*2028*	*990*	*700*	*170*	*1125*

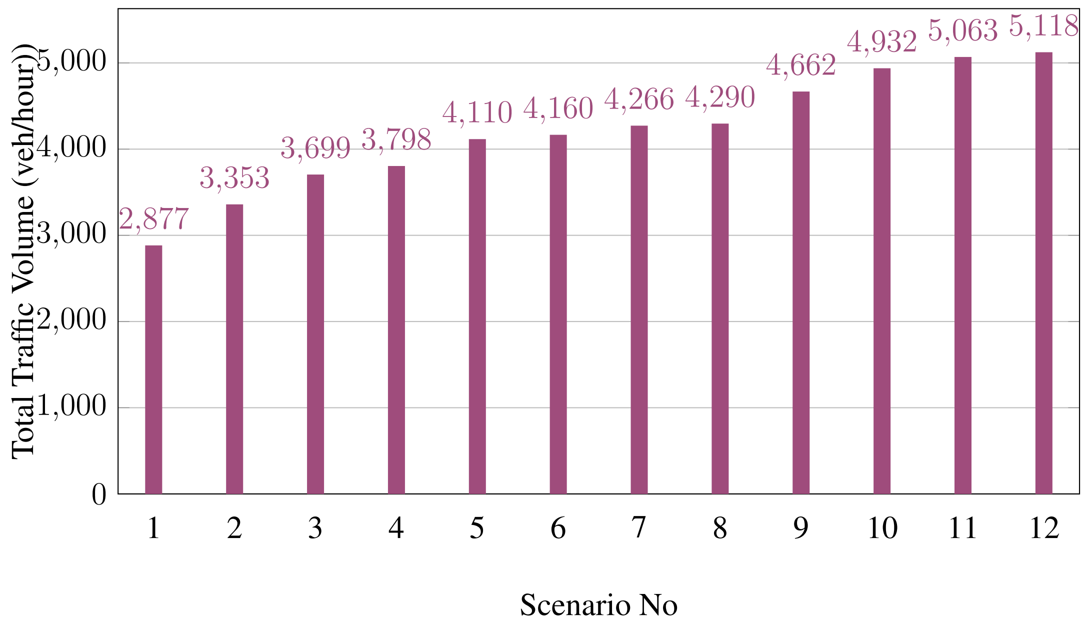


Figure 6.9: Total Traffic Volume of the Intersection 1 For Each Scenario

Finally, in table 6.19, it can be seen the additional informations and assumptions for Intersection 1.

Table 6.19: Assumptions and Informations Of Intersection 2

Assumptions	
Initial Cycle Length	**90 seconds**
Minimum Effective Green Time Per Phase	**8 seconds**
Lost Time Per Phase	**2 seconds**
Yellow Time	**3 Seconds**
All Red Time	**1 Second**
Number Of Phases	**3**
Preassumed Range Value (δ) For NSM	**5**

6.0.2.1 Analyzing Under-Saturated Conditions

In this section, under-saturated conditions that occur according to the determined phase design and scenarios will be examined and a 1-hour delay comparison will be made using the previously mentioned MCLM and CCM. For this intersection, undersaturated conditions occur in Scenario 1 to Scenario 8. For this reason, in this section, delay comparison will be made only between MCLM and CCM.

Optimized effective green times of each phase and **1-hour HCM 2000** average control delay of the intersection with the help of MCLM and CCM can be seen in table 6.20 and table 6.21.

Table 6.20: Optimized Effective Green Times - MCLM - Intersection 2

Scenario No	***Phase-1***	***Phase-2***	***Phase-3***	***Cycle Length (C)***	***1-Hour Average Control Delay (sec/veh)***
1	***8***	***8***	***8***	***33***	***15.38***
2	***8***	***8***	***8***	***33***	**21.79**
3	***10***	***10***	***9***	***38***	**38.16**
4	***10***	***12***	***11***	***42***	***36.77***
5	***16***	***15***	***14***	***54***	***43.16***
6	***18***	***19***	***16***	***62***	***50.91***
7	***21***	***18***	***16***	***64***	**47.76**
8	***21***	***20***	***22***	***72***	**53.79**

Table 6.21: Optimized Effective Green Times - CCM - Intersection 2

Scenario No	***Phase-1***	***Phase-2***	***Phase-3***	***Cycle Length (C)***	***1-Hour Average Control Delay (sec/veh)***
1	***28***	***28***	***25***	***90***	**24.05**
2	***30***	***26***	***25***	***90***	**24.16**
3	***28***	***28***	***25***	***90***	**28.73**
4	***25***	***30***	***26***	***90***	**30.08**
5	***28***	***28***	***25***	***90***	**35.34**
6	***28***	***30***	***23***	***90***	**42.15**
7	***30***	***28***	***23***	***90***	**43.03**
8	***27***	***26***	***28***	***90***	**46.56**

The comparative delay results of **CCM** and **MCLM** according to scenarios can be examined in the bar plot in figure 6.10.

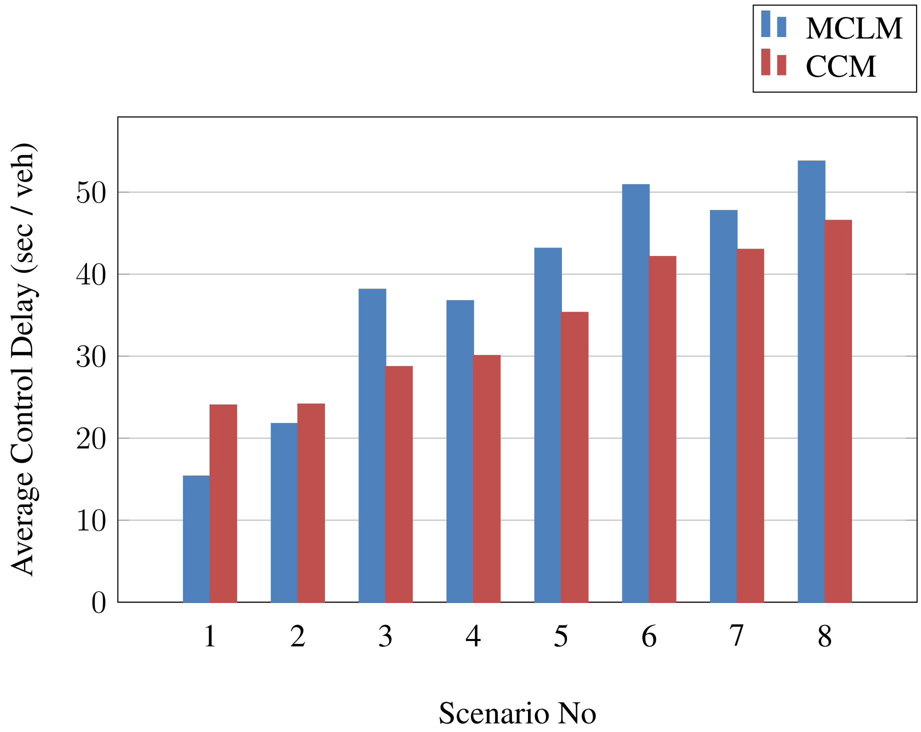


Figure 6.10: 1 Hour Average Control Delay Comparison - Undersaturated Conditions - MCLM and CCM - Intersection 2

As seen in figure 6.10, CCM gave much better delay results compared to CCM in Scenario 3, 4, 5, 6, 7 and 8, whereas MCLM gave less average delay of the intersection compared to CCM in Scenario 1 and 2.

If we examine the delay measurements of each scenario separately;

- ***Scenario 1***
  - **MCLM** - 1-Hour Average Control Delay : **15.38 (sec/veh)**
  - **CCM** - 1-Hour Average Control Delay : **24.05 (sec/veh)**
  - **Delay Improvement Percent** : ***36% by MCLM***
- ***Scenario 2***
  - **MCLM** - 1-Hour Average Control Delay : **21.79 (sec/veh)**

  - **CCM** - 1-Hour Average Control Delay : **24.16 (sec/veh)**
  - **Delay Improvement Percent** : ***9.8% by MCLM***

- ***Scenario 3***
  - **MCLM** - 1-Hour Average Control Delay : **38.16 (sec/veh)**
  - **CCM** - 1-Hour Average Control Delay : **28.73 (sec/veh)**
  - **Delay Improvement Percent** : ***24.7% by CCM***

- ***Scenario 4***
  - **MCLM** - 1-Hour Average Control Delay : **36.77 (sec/veh)**
  - **CCM** - 1-Hour Average Control Delay : **30.08 (sec/veh)**
  - **Delay Improvement Percent** : ***18.1% by CCM***

- ***Scenario 5***
  - **MCLM** - 1-Hour Average Control Delay : **43.16 (sec/veh)**
  - **CCM** - 1-Hour Average Control Delay : **35.34 (sec/veh)**
  - **Delay Improvement Percent** : ***18.1% by CCM***

- ***Scenario 6***
  - **MCLM** - 1-Hour Average Control Delay : **50.91 (sec/veh)**
  - **CCM** - 1-Hour Average Control Delay : **42.15 (sec/veh)**
  - **Delay Improvement Percent** : ***17.2% by CCM***

- ***Scenario 7***
  - **MCLM** - 1-Hour Average Control Delay : **47.76 (sec/veh)**
  - **CCM** - 1-Hour Average Control Delay : **43.03 (sec/veh)**
  - **Delay Improvement Percent** : ***9.9% by CCM***

- ***Scenario 8***
  - **MCLM** - 1-Hour Average Control Delay : **53.79 (sec/veh)**

- **CCM** - 1-Hour Average Control Delay : **46.56 (sec/veh)**
- **Delay Improvement Percent** : ***13.4% by CCM***

As seen from these measurements and delay improvement percentages, using effective green times of CCM for these scenarios would be more appropriate than MCLM in this intersection. The reason MCLM gives a higher delay at the end of 1 hour compared to CCM is that MCLM causes higher degrees of saturation in some lane groups (see chapter 5).

6.0.2.2 Analyzing Over-Saturated Conditions

In this intersection, over-saturated conditions occur in Scenario 8 up to Scenario 12. For this reason, in this section, delay comparison between MTQLM, MMQLM and NSM will be made for only for these scenarios.

Optimized effective green times of each phase and **1-hour HCM 2000** average control delay of the intersection with the help of MTQLM, MMQLM and NSM can be seen table 6.22, table 6.23 and table 6.24.

Table 6.22: Optimized Effective Green Times - MTQLM - Intersection 2

Scenario No	*Phase-1*	*Phase-2*	*Phase-3*	*Cycle Length (C)*	*1-Hour Average Control Delay (sec/veh)*
9	***31***	***27***	***23***	***90***	**89.63**
10	***31***	***30***	***20***	***90***	**191.63**
11	***33***	***30***	***18***	***90***	**322.46**
12	***33***	***35***	***13***	***90***	**522.97**

Table 6.23: Optimized Effective Green Times - MMQLM - Intersection 2

Scenario No	*Phase-1*	*Phase-2*	*Phase-3*	*Cycle Length (C)*	*1-Hour Average Control Delay (sec/veh)*
9	***30***	***27***	***24***	***90***	**93.33**
10	***29***	***27***	***25***	***90***	**146.05**
11	***29***	***26***	***26***	***90***	**238.96**
12	***28***	***29***	***24***	***90***	**300.07**

Table 6.24: Optimized Effective Green Times - NSM - Intersection 2

Scenario No	*Phase-1*	*Phase-2*	*Phase-3*	*Cycle Length (C)*	*1-Hour Average Control Delay (sec/veh)*
9	***32***	***24***	***25***	***90***	**78.88**
10	***31***	***24***	***26***	***90***	**122.55**
11	***33***	***21***	***27***	***90***	**184.97**
12	***33***	***24***	***24***	***90***	**237.62**

The comparative delay results of MTQLM, MMQLM and NSM according to scenarios can be examined in the bar plot in figure 6.11.

As seen from the delay measurements of each method, as in Intersection 1, NSM always gave much better delay results compared to other 2 methods in every scenario. In addition, As the total of hourly traffic volume of the intersection in every lane group, MTQLM always gave higher delay compared to MMQLM and NSM. The reason for that is that increase in unfairness in the residual queue allocation in the MTQLM as the traffic volume of intersection increases, as in Intersection 1 (see chapter 5).

If we examine the delay measurements of each scenario separately;

- ***Scenario 9***
  - **MTQLM** - 1-Hour Average Control Delay : **89.63 (sec/veh)**
  - **MMQLM** - 1-Hour Average Control Delay : **93.33 (sec/veh)**

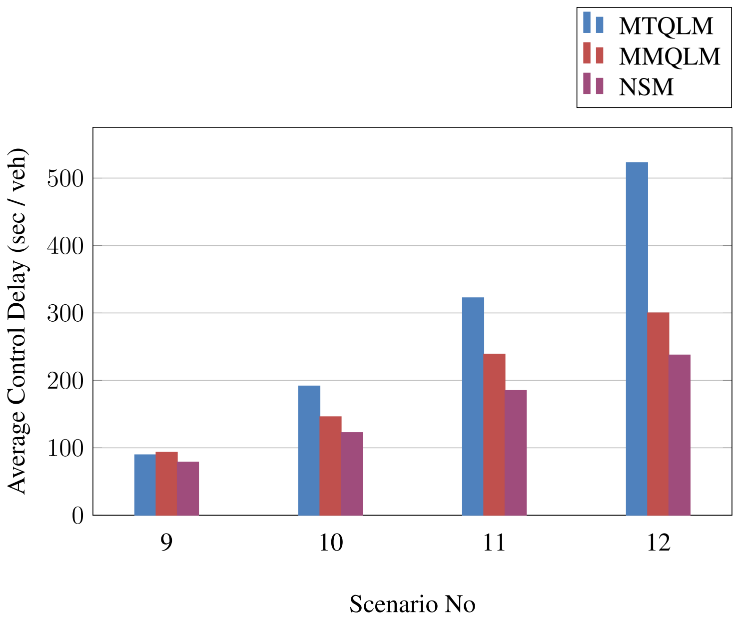


Figure 6.11: 1 Hour Average Control Delay Comparison - Oversaturated Conditions - MTQLM, MMQLM and NSM - Intersection 2

- **NSM** - 1-Hour Average Control Delay : **78.88 (sec/veh)**
- (**First Stage**) **Delay Improvement Percent** : ***4% by MTQLM***
- (**Second Stage**) **Delay Improvement Percent** : ***12% by NSM***

- ***Scenario 10***
  - **MTQLM** - 1-Hour Average Control Delay : **191.63 (sec/veh)**
  - **MMQLM** - 1-Hour Average Control Delay : **146.05 (sec/veh)**
  - **NSM** - 1-Hour Average Control Delay : **122.55 (sec/veh)**
  - (**First Stage**) **Delay Improvement Percent** : ***23.8% by MMQLM***
  - (**Second Stage**) **Delay Improvement Percent** : ***16.0% by NSM***

- ***Scenario 11***
  - **MTQLM** - 1-Hour Average Control Delay : **322.46 (sec/veh)**
  - **MMQLM** - 1-Hour Average Control Delay : **238.96 (sec/veh)**
  - **NSM** - 1-Hour Average Control Delay : **184.97 (sec/veh)**
  - (**First Stage**) **Delay Improvement Percent** : ***25.9% by MMQLM***

– **(Second Stage) Delay Improvement Percent** : ***22.6% by NSM***

- ***Scenario 12***

  – **MTQLM** - 1-Hour Average Control Delay : **522.97 (sec/veh)**

  – **MMQLM** - 1-Hour Average Control Delay : **300.07 (sec/veh)**

  – **NSM** - 1-Hour Average Control Delay : **237.62 (sec/veh)**

  – **(First Stage) Delay Improvement Percent** : ***42.6% by MMQLM***

  – **(Second Stage) Delay Improvement Percent** : ***20.8% by NSM***

Finally, delay comparison results of exhaustive search algorithm and NSM can be seen in table 6.25. According to the results in the table 6.25, NSM could find the global minimum in Scenario 9, 10 and 11, while it could not find the global minimum in Scenario 12.

Table 6.25: Delay Comparison Between Exhaustive Search and NSM - Intersection 2

Scenario No	***1-Hour Delay (sec/veh) NSM***	***1-Hour Delay (sec/veh) Exhaustive Search***	***Is Global Minimum Found ?***
7	***78.88***	***78.88***	**Yes**
8	***122.55***	***122.55***	**Yes**
9	***184.97***	***184.97***	**Yes**
10	***237.62***	***235.29***	**No**

6.0.2.3 Preferred Methods for Intersection 2

According to the delay comparisons made in the previous sections, the methods to be preferred for each scenario and effective green times can be seen in table 6.26.

Table 6.26: Preferred Methods - Intersection 2

Scenario No	*Phase-1*	*Phase-2*	*Phase-3*	*Cycle Length (C)*	*Preferred Method*	*1-Hour Delay (sec/veh)*
1	*8*	*8*	*8*	*33*	*MCLM*	*15.38*
2	*8*	*8*	*8*	*33*	*MCLM*	*21.79*
3	*28*	*28*	*25*	*90*	*CCM*	*28.73*
4	*25*	*30*	*26*	*90*	*CCM*	*30.08*
5	*28*	*28*	*25*	*90*	*CCM*	*35.34*
6	*28*	*30*	*23*	*90*	*CCM*	*42.15*
7	*28*	*30*	*23*	*90*	*CCM*	*43.03*
8	*27*	*26*	*28*	*90*	*CCM*	*46.56*
9	*32*	*24*	*25*	*90*	*NSM*	*78.88*
10	*31*	*24*	*26*	*90*	*NSM*	*122.55*
11	*33*	*21*	*27*	*90*	*NSM*	*184.97*
12	*33*	*24*	*24*	*90*	*NSM*	*237.62*

6.0.3 Experiment 3 - Analyzing Intersection 3

Intersection 3 is a type of 4-legged intersection like **Intersection 1**. It has additional lanes for all left, right, straight movements. In this intersection, all right turns gain right of way in 2 consecutive phases. The intersection geometry can be seen in figure 6.12.

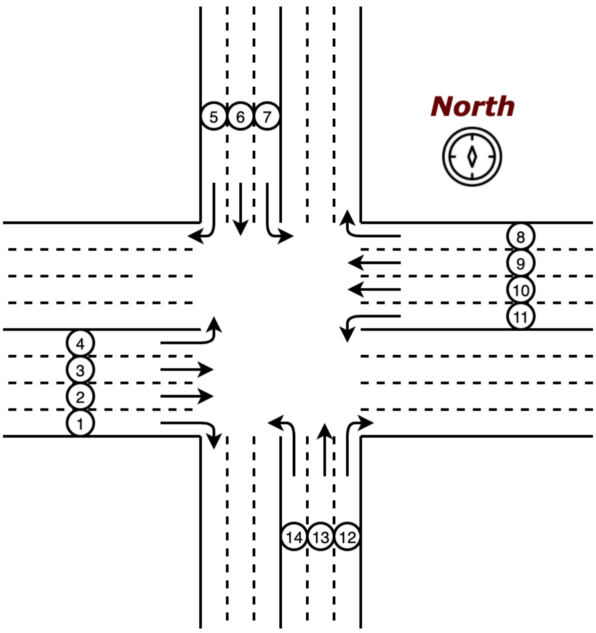


Figure 6.12: Intersection 3 - Geometry

As seen in figure 6.12, Intersection 3 has a **total of 14 lanes**. According to the movements in lanes, these lanes combine to form **12 different lane groups**. These lane groups for Intersection 3 can be seen in figure 6.13.

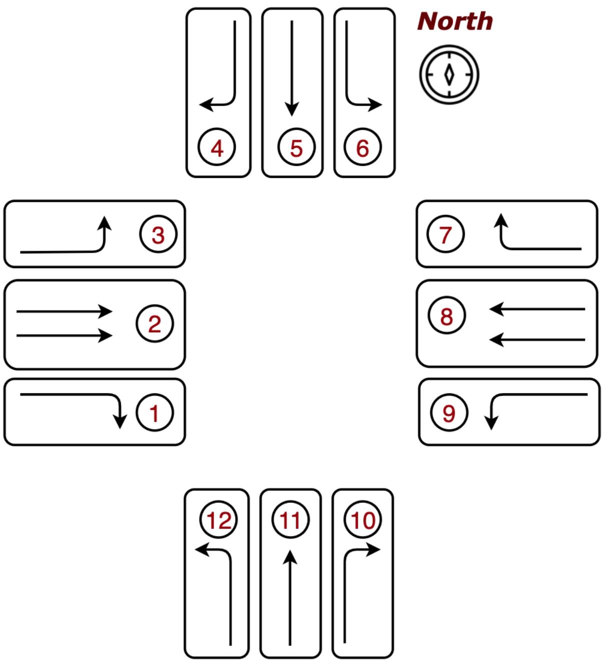


Figure 6.13: Intersection 3 - Lane Groups

Table 6.27: Lane Groups and Namings - Intersection 3

Lane Group No	*Naming*	*Number Of Lanes*
1	***west-right or w-r***	***1***
2	***west-straight or w-s***	***2***
3	***west-left or w-l***	***1***
4	***north-right or n-r***	***1***
5	***north-straight or n-s***	***1***
6	***north-left or n-l***	***1***
7	***east-right or e-r***	***1***
8	***east-straight or e-s***	***2***
9	***east-left or e-l***	***1***
10	***south-right or s-r***	***1***
11	***south-straight or s-s***	***1***
12	***south-left or s-l***	***1***

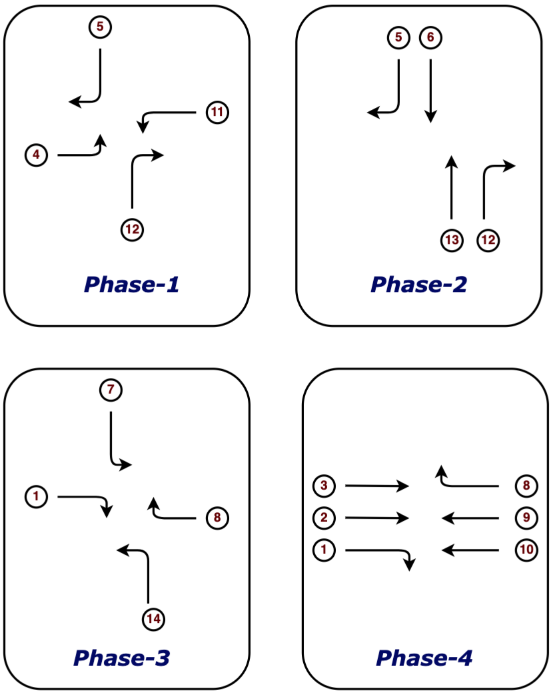


Figure 6.14: Intersection 3 - Phase Design-1

For this intersection, unlike Intersections 1 and 2, two possible phase designs were created in order to show the effect of phase design on average control delay of the intersection. The first phase design and the movements involved in each phase can be seen in figure 6.14.

In table 6.28, lane groups and the phases they gain right of way in the first phase design, **Phase Design 1**, can be seen.

The second phase design and the movements involved in each phase can be seen in figure 6.15.

In table 6.29, lane groups and the phases they gain right of way in the second phase design, **Phase Design 2**, can be seen.

Table 6.28: Phase Design 1 and Lane Groups - Right Of Way - Intersection 3

Lane Group No	*Right Of Way - Phases*
1	*Phase-3 and Phase-4*
2	*Phase-4*
3	*Phase-1*
4	*Phase-1 and Phase-2*
5	*Phase-2*
6	*Phase-3*
7	*Phase-3 and Phase-4*
8	*Phase-4*
9	*Phase-1*
10	*Phase-1 and Phase-2*
11	*Phase-2*
12	*Phase-3*

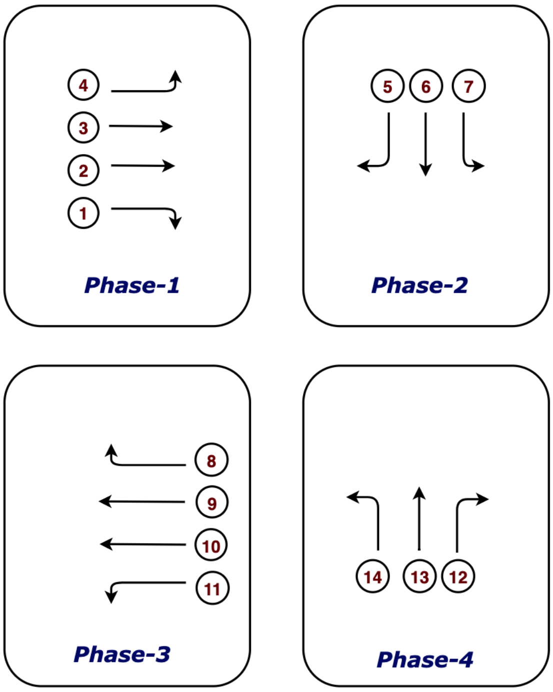


Figure 6.15: Intersection 3 - Phase Design-2

Table 6.29: Phase Design 2 and Lane Groups - Right Of Way - Intersection 3

Lane Group No	***Right Of Way - Phases***
1	***Phase-1***
2	***Phase-1***
3	***Phase-1***
4	***Phase-2***
5	***Phase-2***
6	***Phase-2***
7	***Phase-3***
8	***Phase-3***
9	***Phase-3***
10	***Phase-4***
11	***Phase-4***
12	***Phase-4***

For this intersection, **12 different traffic scenarios** were generated. Hourly traffic volumes of the base scenario by **direction** can be seen in table 6.30.

Table 6.30: Hourly Traffic Volumes(veh/hour) By Direction - Base Scenario - Intersection 3

	Directions		
	Left	***Straight***	***Right***
West	*150 (veh/hour)*	*500 (veh/hour)*	*100 (veh/hour)*
North	*250 (veh/hour)*	*450 (veh/hour)*	*150 (veh/hour)*
East	*350 (veh/hour)*	*800 (veh/hour)*	*125 (veh/hour)*
South	*150 (veh/hour)*	*150 (veh/hour)*	*100 (veh/hour)*

Hourly traffic volumes of each lane group in each scenario can be seen in table 6.31.

Table 6.31: Hourly Traffic Volumes (veh/hour) By Lane Group For Each Scenario -Intersection 3

	Lane Groups											
Scenario No	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	60	300	90	90	270	150	75	480	210	60	90	90
2	70	350	105	105	270	150	88	560	245	70	105	105
3	80	400	120	120	360	225	100	640	350	80	120	120
4	80	400	120	120	450	225	100	640	315	80	120	120
5	100	450	135	150	450	250	125	720	315	100	150	150
6 (Base Scenario)	100	500	150	150	450	250	125	800	350	100	150	150
7	110	550	165	165	495	275	138	880	350	110	165	165
8	110	550	165	165	495	275	138	1040	385	110	165	165
9	110	550	165	165	585	288	138	1040	420	110	165	165
10	130	650	195	195	585	325	163	1040	455	130	195	195
11	130	650	195	195	585	325	163	1040	455	130	300	195
12	130	720	195	195	585	325	163	1040	455	130	300	195

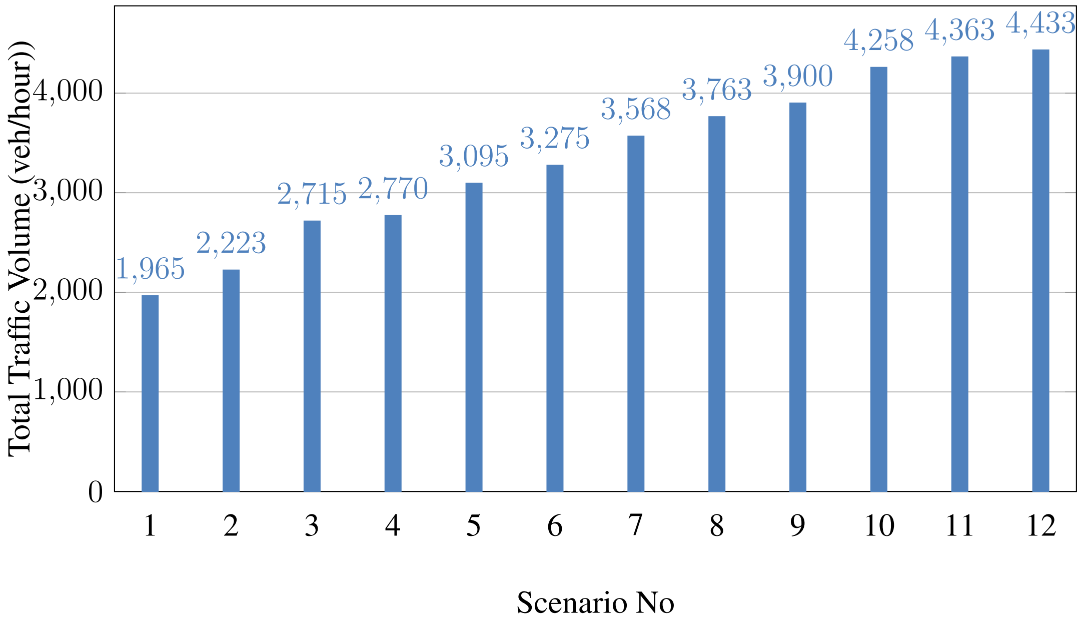


Figure 6.16: Total Traffic Volume of the Intersection 1 For Each Scenario

Finally, in table 6.32, it can be seen the additional informations and assumptions for Intersection 3.

Table 6.32: Assumptions and Informations Of Intersection 3

Assumptions	
Initial Cycle Length	**135 seconds**
Minimum Effective Green Time Per Phase	**9 seconds**
Lost Time Per Phase	**2 seconds**
Yellow Time	**3 Seconds**
All Red Time	**1 Second**
Number Of Phases	**4**
Preassumed Range Value (δ) For NSM	**5**

For this intersection, the delay comparison of each method has been made and the methods to be preferred for each scenario are determined. For this intersection, the delay comparisons between methods will not be discussed for sake of simplicity.

6.0.3.1 Preferred Methods - Intersection 3

In this section, preferred methods for Intersection 3, using 2 separate phase designs, Phase Design 1 and Phase Design 2, are presented. In addition, delay comparisons between these two phase designs were made and the reasons in difference between delays are discussed.

In table 6.33, it can be examined the preferred methods and effective green times when **Phase Design 1** is used.

Table 6.33: Preferred Methods - Phase Design 1 - Intersection 3

Scenario No	*Phase-1*	*Phase-2*	*Phase-3*	*Phase-4*	*Cycle Length (C)*	*Preferred Method*	*1-Hour Delay (sec/veh)*
1	*9*	*9*	*9*	*9*	*48*	**MCLM**	**22.58**
2	*9*	*9*	*9*	*9*	*48*	**MCLM**	**24.75**
3	*31*	*36*	*32*	*24*	*135*	**CCM**	**50.49**
4	*26*	*44*	*31*	*22*	*135*	**CCM**	**53.22**
5	*11*	*15*	*12*	*9*	*59*	**MCLM**	**53.84**
6	*14*	*18*	*16*	*10*	*70*	**MCLM**	**58.29**
7	*26*	*40*	*35*	*22*	*135*	**CCM**	**71.37**
8	*28*	*35*	*40*	*20*	*135*	**NSM**	**99.37**
9	*27*	*38*	*39*	*19*	*135*	**NSM**	**163.14**
10	*28*	*37*	*38*	*20*	*135*	**NSM**	**193.96**
11	*28*	*37*	*38*	*20*	*135*	**NSM**	**190.46**
12	*28*	*37*	*38*	*20*	*135*	**NSM**	**190.72**

In table 6.34, it can be examined the preferred methods and effective green times when **Phase Design 2** is used.

Table 6.34: Preferred Methods - Phase Design 2 - Intersection 3

Scenario No	*Phase-1*	*Phase-2*	*Phase-3*	*Phase-4*	*Cycle Length (C)*	*Preferred Method*	*1-Hour Delay (sec/veh)*
1	*9*	*9*	*9*	*9*	*48*	**MCLM**	**24.00**
2	*9*	*9*	*9*	*9*	*48*	**MCLM**	**26.28**
3	*9*	*10*	*10*	*9*	*50*	**MCLM**	**47.12**
4	*9*	*14*	*10*	*9*	*54*	**MCLM**	**47.40**
5	*24*	*45*	*38*	*16*	*135*	**CCM**	**51.27**
6	*9*	*15*	*13*	*9*	*58*	**MCLM**	**51.55**
7	*25*	*42*	*40*	*16*	*135*	**CCM**	**56.97**
8	*24*	*41*	*44*	*14*	*135*	**CCM**	**61.69**
9	*22*	*45*	*42*	*14*	*135*	**CCM**	**71.44**
10	*25*	*44*	*39*	*15*	*135*	**CCM**	**92.54**
11	*24*	*40*	*39*	*20*	*135*	**NSM**	**125.90**
12	*27*	*38*	*39*	*19*	*135*	**NSM**	**142.36**

If we compare the 1-hour average control delay with effective green times obtained using 2 different phase designs for each scenario;

- ***Scenario 1***
  - **Phase Design 1** - 1-Hour Average Control Delay : **22.58 (sec/veh)**
  - **Phase Design 2** - 1-Hour Average Control Delay : **24.00 (sec/veh)**
  - **Delay Improvement Percent** : ***5.92% by Phase Design 1***

- ***Scenario 2***
  - **Phase Design 1** - 1-Hour Average Control Delay : **24.75 (sec/veh)**
  - **Phase Design 2** - 1-Hour Average Control Delay : **26.28 (sec/veh)**
  - **Delay Improvement Percent** : ***5.82% by Phase Design 1***
- ***Scenario 3***
  - **Phase Design 1** - 1-Hour Average Control Delay : **50.49 (sec/veh)**
  - **Phase Design 2** - 1-Hour Average Control Delay : **47.12 (sec/veh)**
  - **Delay Improvement Percent** : ***6.67% by Phase Design 2***
- ***Scenario 4***
  - **Phase Design 1** - 1-Hour Average Control Delay : **53.22 (sec/veh)**
  - **Phase Design 2** - 1-Hour Average Control Delay : **47.40 (sec/veh)**
  - **Delay Improvement Percent** : ***10.9% by Phase Design 2***
- ***Scenario 5***
  - **Phase Design 1** - 1-Hour Average Control Delay : **53.84 (sec/veh)**
  - **Phase Design 2** - 1-Hour Average Control Delay : **51.27 (sec/veh)**
  - **Delay Improvement Percent** : ***4.8% by Phase Design 2***
- ***Scenario 6***
  - **Phase Design 1** - 1-Hour Average Control Delay : **58.29 (sec/veh)**
  - **Phase Design 2** - 1-Hour Average Control Delay : **51.55 (sec/veh)**
  - **Delay Improvement Percent** : ***11.6% by Phase Design 2***
- ***Scenario 7***
  - **Phase Design 1** - 1-Hour Average Control Delay : **71.37 (sec/veh)**
  - **Phase Design 2** - 1-Hour Average Control Delay : **56.97 (sec/veh)**
  - **Delay Improvement Percent** : ***20.2% by Phase Design 2***

- *Scenario 8*
  - **Phase Design 1** - 1-Hour Average Control Delay : **99.37 (sec/veh)**
  - **Phase Design 2** - 1-Hour Average Control Delay : **61.69 (sec/veh)**
  - **Delay Improvement Percent** : ***37.9% by Phase Design 2***
- *Scenario 9*
  - **Phase Design 1** - 1-Hour Average Control Delay : **163.14 (sec/veh)**
  - **Phase Design 2** - 1-Hour Average Control Delay : **71.44 (sec/veh)**
  - **Delay Improvement Percent** : ***56.2% by Phase Design 2***
- *Scenario 10*
  - **Phase Design 1** - 1-Hour Average Control Delay : **193.96 (sec/veh)**
  - **Phase Design 2** - 1-Hour Average Control Delay : **92.54 (sec/veh)**
  - **Delay Improvement Percent** : ***52.3% by Phase Design 2***
- *Scenario 11*
  - **Phase Design 1** - 1-Hour Average Control Delay : **190.46 (sec/veh)**
  - **Phase Design 2** - 1-Hour Average Control Delay : **125.90 (sec/veh)**
  - **Delay Improvement Percent** : ***33.9% by Phase Design 2***
- *Scenario 12*
  - **Phase Design 1** - 1-Hour Average Control Delay : **190.72 (sec/veh)**
  - **Phase Design 2** - 1-Hour Average Control Delay : **142.36 (sec/veh)**
  - **Delay Improvement Percent** : ***25.4% by Phase Design 2***

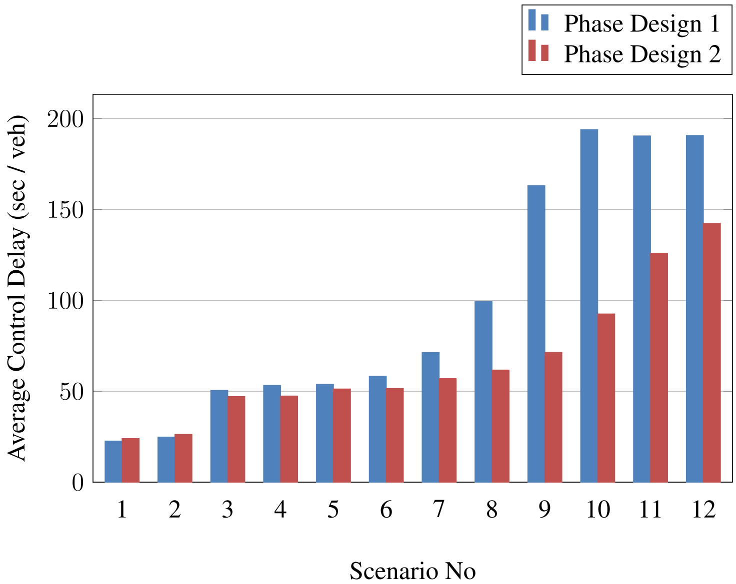


Figure 6.17: Delay Comparison - Phase Design 1 and Phase Design 2 - Intersection 3

As seen in figure 6.17, the delays caused by using Phase Design 1 are always higher than when Phase Design 2 is used, except for Scenario 1 and Scenario 2. While the delay differences were low until Scenario 6, the average control delay differences gradually increased after the 6th scenario. For example, using **Phase Design 2** in scenario 9 reduced the delay by **56.2** percent compared to the **Phase Design 1**.

Table 6.35: Preferred Phase Designs and Effective Green Times For Each Scenario - Intersection 3

Scenario No	*Phase-1*	*Phase-2*	*Phase-3*	*Phase-4*	*Cycle Length (C)*	*Preferred Phase Design*	*1-Hour Delay (sec/veh)*
1	*9*	*9*	*9*	*9*	*48*	**1**	**22.58**
2	*9*	*9*	*9*	*9*	*48*	**1**	**24.75**
3	*9*	*10*	*10*	*9*	*50*	**2**	**47.12**
4	*9*	*14*	*10*	*9*	*54*	**2**	**47.40**
5	*24*	*45*	*38*	*16*	*135*	**2**	**51.27**
6	*9*	*15*	*13*	*9*	*58*	**2**	**51.55**
7	*25*	*42*	*40*	*16*	*135*	**2**	**56.97**
8	*24*	*41*	*44*	*14*	*135*	**2**	**61.69**
9	*22*	*45*	*42*	*14*	*135*	**2**	**71.44**
10	*25*	*44*	*39*	*15*	*135*	**2**	**92.54**
11	*24*	*40*	*39*	*20*	*135*	**2**	**125.90**
12	*27*	*38*	*39*	*19*	*135*	**2**	**142.36**

Finally, the phase design and effective green times minimizing average delay for Intersection 3 can be seen in table 6.35.

6.0.4 Testing Intersection 3 on PTV VISSIM - Calibration and Simulation

In this section, Intersection 3, which was previously evaluated with the HCM 2000 delay calculation, will be tested in the PTV VISSIM simulation environment. The purpose of the simulation here is to calibrate intersection geometry, vehicle speeds, driver behavior and vehicle composition parameters, and to try to obtain delay results close to the delays previously obtained by analytical method (HCM 2000) in the PTV VISSIM environment.

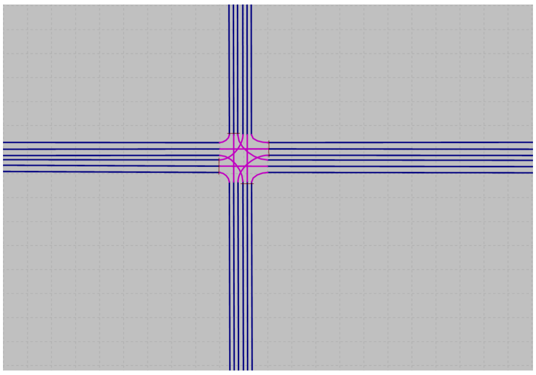
PTV VISSIM has a stochastic nature and uses the Poisson distribution for when vehicles enter the network. Therefore, it can be predicted that there will be a certain difference between the average vehicle delay measured in PTV VISSIM and the delay obtained from analytical delay methods (HCM 2000). However, in this study, these delay differences were aimed to be as small as possible.

In the PTV VISSIM environment, the saturation flow rates can not be directly given as a value for simulation. Therefore, PTV VISSIM reflects the saturation flow rate using driver behavior and car-following logic. Within the scope of this study, by changing the driver behavior parameters, the lane-based saturation flow rate was tried to be approached to the predetermined 1800 veh / hour / lane. However, it should be known that it is almost impossible for this saturation flow rate to be exactly 1800 veh / hour / lane in the simulation.

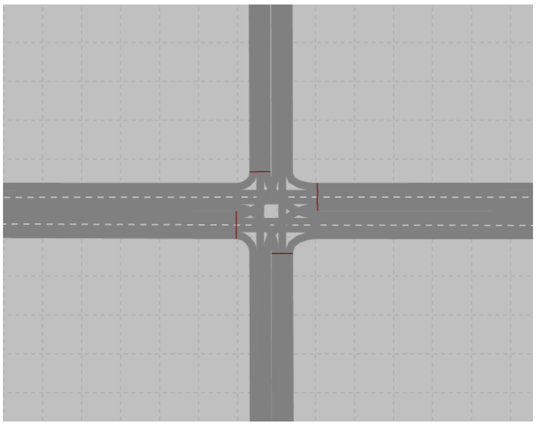
In the following sections, the reflection of the intersection geometry in the simulation environment will be explained step by step.

6.0.4.1 Intersection Geometry,Vehicle Inputs, Routes and Vehicle Compositions

First, in the PTV VISSIM environment, links and connectors between links were used to create intersection geometry.



(a) Links - Wireframe



(b) Links - Complete

Figure 6.18: PTV VISSIM - Intersection Geometry and Links (Lanes) - Intersection 3

The next step after creating the links and connectors between links is to give hourly vehicle volumes as vehicle input to the links here for Intersection 3. At this step, Scenario 6 has been selected as the sample scenario. The hourly volumes of each lane group in Scenario 6 can be seen in figure 6.19.

All vehicles in the network can make right movements, left movements and straight movements according to their lanes. For Intersection 3, there is only one movement (left, right, straight) from each lane. For this reason, vehicle routing feature was not used for this intersection.

In VISSIM environment, bus, heavy vehicles (HGV), SUVs, pedestrians, motorcy-

Vehicle Inputs / Vehicle Volumes By Time Interval

Select layout... Vehicle volumes by

Count: 12	No	Name	Link	Volume(0)	VehComp(0)
1	1		3: West-Left	150,0	1: Only Cars
2	2		1: West-Through	500,0	1: Only Cars
3	3		2: West-Right	100,0	1: Only Cars
4	4		15: North-Left	250,0	1: Only Cars
5	5		16: North-Throu...	450,0	1: Only Cars
6	6		17: North-Right	150,0	1: Only Cars
7	9		28: East-Right	125,0	1: Only Cars
8	10		5: South-Left	150,0	1: Only Cars
9	11		6: South-Through	150,0	1: Only Cars
10	12		7: South-Right	100,0	1: Only Cars
11	13		25: East-Left	350,0	1: Only Cars
12	14		26: East-Througl	800,0	1: Only Cars

Figure 6.19: PTV VISSIM - Vehicle Inputs - Scenario 6 (Sample Scenario)

clists and bicyclists were not taken into account. Therefore, all vehicles in the network are passenger cars.

The desired speeds of passenger cars in the network were tested at 50 km/h, 40 km/h and 30 km/h, respectively. As a result of the measurements (delay measurements), it was decided that the reasonably desired speed was 30 km/h.

Vehicle compositions and their desired speeds created in VISSIM can be seen in figure 6.20.

6.0.4.2 Signal Heads,Signal Controller and Phase Design

In order to set the traffic signal timings, first, signal heads have been placed in each lane group.

After this step, a 4-phase signal controller was created. As previously assumed, 3 seconds of yellow time (amber time) after each phase actual green time and 1 second of all-red time were added on the signal controller.

Finally, the actual green times of each phase that need to be noted. As mentioned earlier, effective green times of each phase are 1 second longer than actual green

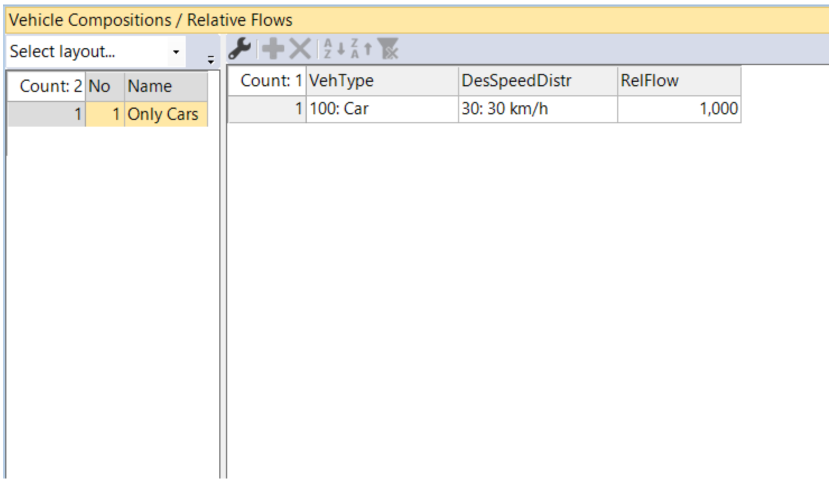


Figure 6.20: PTV VISSIM - Vehicle Compositions and Desired Speeds - Scenario 6 (Sample Scenario)

time, according to the assumptions made (lost time per phase).

The phase design and times, actual green time, yellow time and all red time, of a sample scenario (Scenario 6) can be seen in figure 6.21.

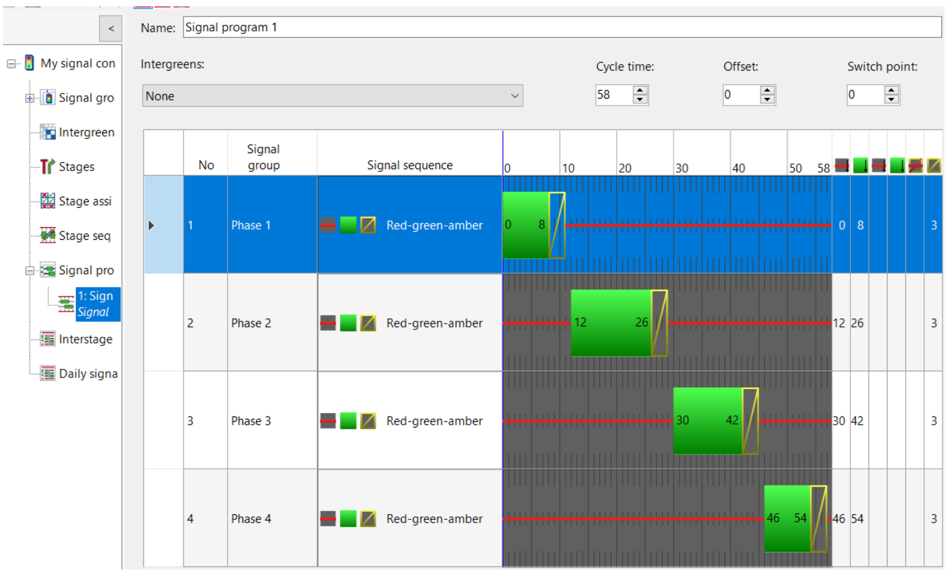


Figure 6.21: PTV VISSIM - Phase Design - Scenario 6 (Sample Scenario)

After all these steps, the 2-D and 3-D views of Intersection 3 while the simulation is running (at any moment) can be seen in figure 6.22 and figure 6.23.

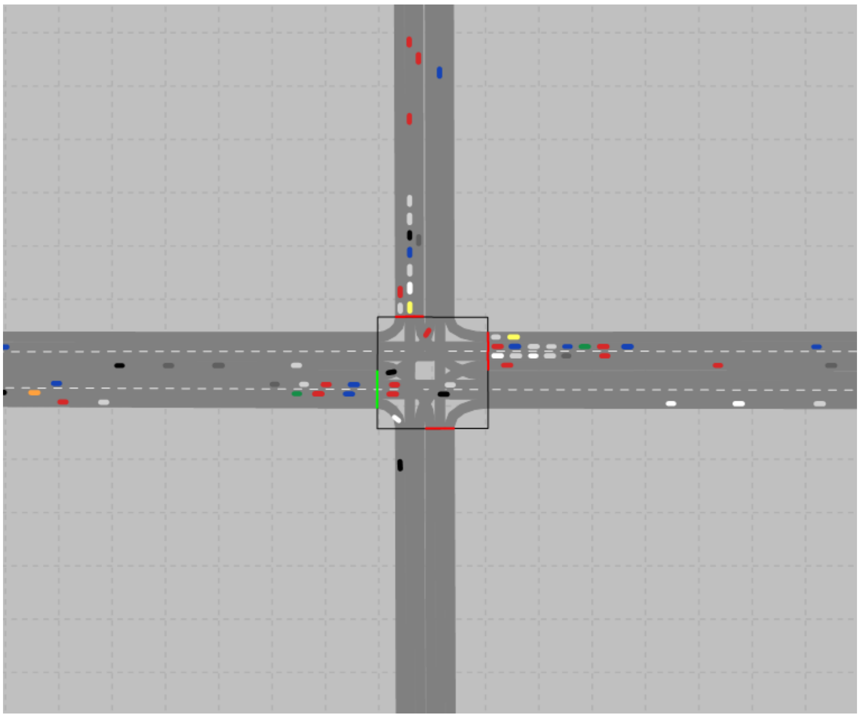


Figure 6.22: PTV VISSIM - Simulation - 2D View

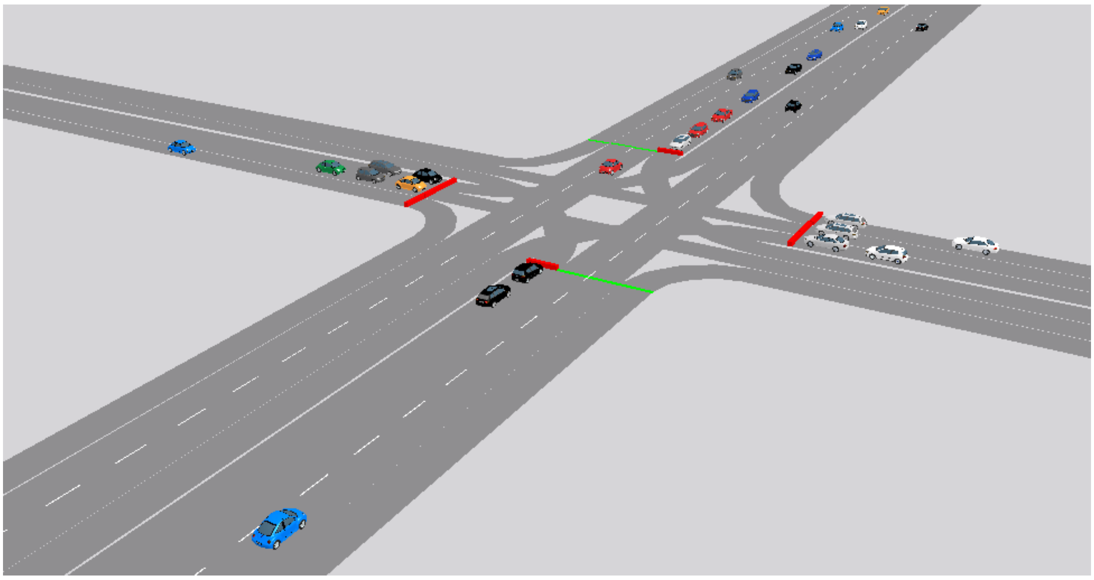


Figure 6.23: PTV VISSIM - Simulation - 3D View

6.0.4.3 Node Evaluations and Adjusting Driver Behaviours

In VISSIM simulation environment, nodes are used to perform and evaluate measurements such as average vehicle delays, fuel consumption, average queue length (m). For this reason, for Intersection 3, a node area was created with the boundaries of the stop lines (signal heads) of each direction. This node area can be seen in the figure. Average vehicle delay measurements were made using this node area (node evaluation).

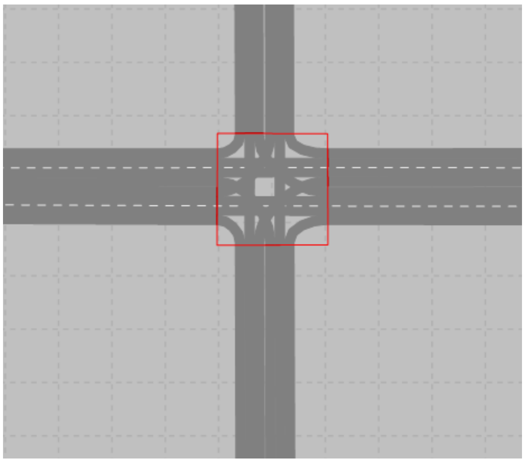


Figure 6.24: PTV VISSIM - Node Area

It is necessary to pay attention to one more point when making delay measurements using this node area. When measuring the average vehicle delay, the parameter of how many meters behind the stop line will the delay be measured is very important. This parameter can be adjusted by changing the delay segment value in VISSIM. As a result of the research in the literature, it was deemed appropriate to set this value as 100 meters behind stop lines. However, especially in scenarios with high traffic volume, this parameter has been updated to 200,300 and 400 meters in some scenarios, as queues longer than 100 meters may occur.

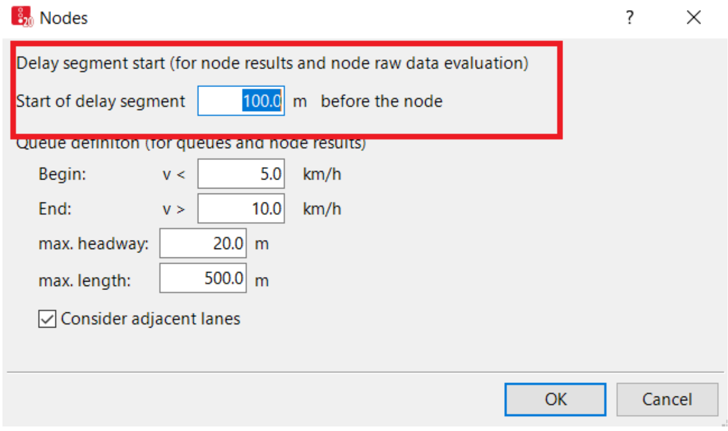
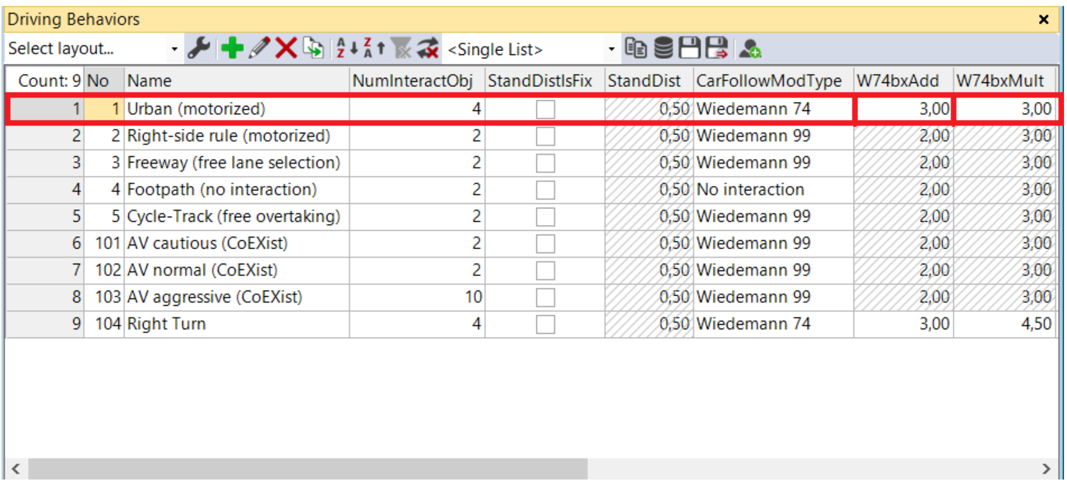


Figure 6.25: PTV VISSIM - Node Evaluation and Delay Segment

Finally, for this intersection, the lane-based saturation flow rate was tried to be approached to the predetermined 1800 veh / hour / lane by changing the driver behavior parameters. VISSIM uses the Wiedemann 74 car following algorithm by default at intersections. In this study, the **w74bxAdd and w74bxMult** coefficients of this algorithm were changed and the delay results were monitored. In Vissim, by default, the values of these coefficients are 2 for w74bxAdd and 3 for w74bxMult, respectively. With default values, Vissim measured the average vehicle delays much lower than the HCM 2000 delay results. For this reason, these coefficients have been changed many times by trial-and-error method and it was decided that the most suitable value is 3 for w74bxAdd and 3 for w74bxMult (Figure 6.26).



Count: 9	No	Name	NumInteractObj	StandDistIsFix	StandDist	CarFollowModType	W74bxAdd	W74bxMult
1	1	Urban (motorized)	4		0,50	Wiedemann 74	3,00	3,00
2	2	Right-side rule (motorized)	2		0,50	Wiedemann 99	2,00	3,00
3	3	Freeway (free lane selection)	2		0,50	Wiedemann 99	2,00	3,00
4	4	Footpath (no interaction)	2		0,50	No interaction	2,00	3,00
5	5	Cycle-Track (free overtaking)	2		0,50	Wiedemann 99	2,00	3,00
6	101	AV cautious (CoEXist)	2		0,50	Wiedemann 99	2,00	3,00
7	102	AV normal (CoEXist)	2		0,50	Wiedemann 99	2,00	3,00
8	103	AV aggressive (CoEXist)	10		0,50	Wiedemann 99	2,00	3,00
9	104	Right Turn	4		0,50	Wiedemann 74	3,00	4,50

Figure 6.26: PTV VISSIM - Driving Behaviours and Parameters

6.0.4.4 Simulation Parameters

In this study, simulation was run for 1 hour for each scenario in VISSIM environment. However, running the simulation only once with one random seed value (42 by default) can be misleading because of the stochastic nature of VISSIM in terms of delays. For this reason, the simulation has to be run several times with different random seed values and averaged the delay in each random seed value. However, it is not entirely clear how many times the simulation has to be run. For this reason, in this study, firstly the simulation was run 2 times and starting from 42, the random seed value was increased by 2 in each simulation run. Then, the delay in each simulation was averaged and noted. This experiment was carried out until the simulation was run 20 times and each average value was noted. These values can be seen in figure 6.27. Scenario 6 was used for this experiment.

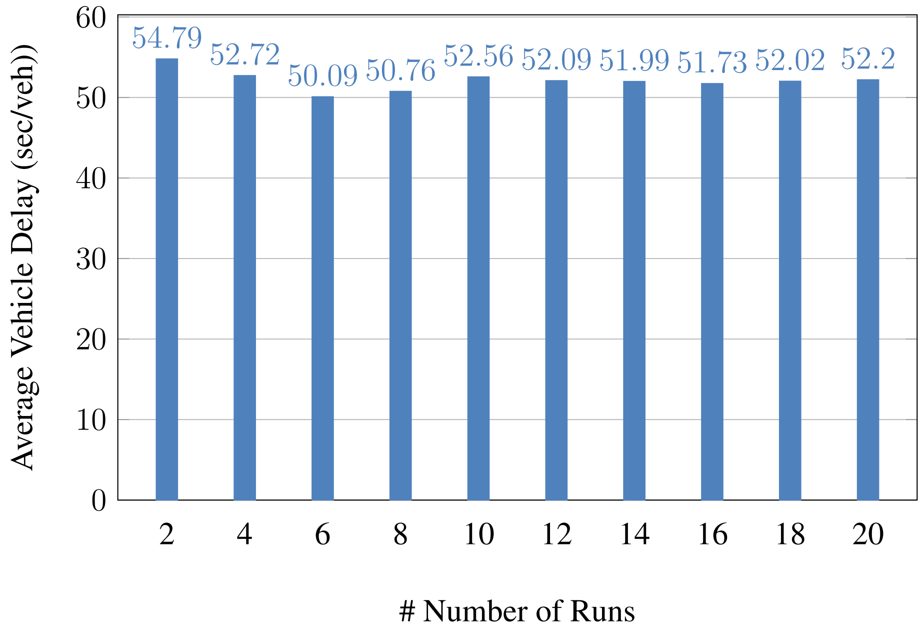


Figure 6.27: Scenario 6 - Delay results with different number of simulation runs at Intersection 3

As seen in figure 6.27, the average vehicle delay results of the intersection are almost the same in the number of runs 10, 12, 14, 16, 18, 20. Based on this, it was found appropriate to run the simulation 10 times for each scenario (Figure 6.28).

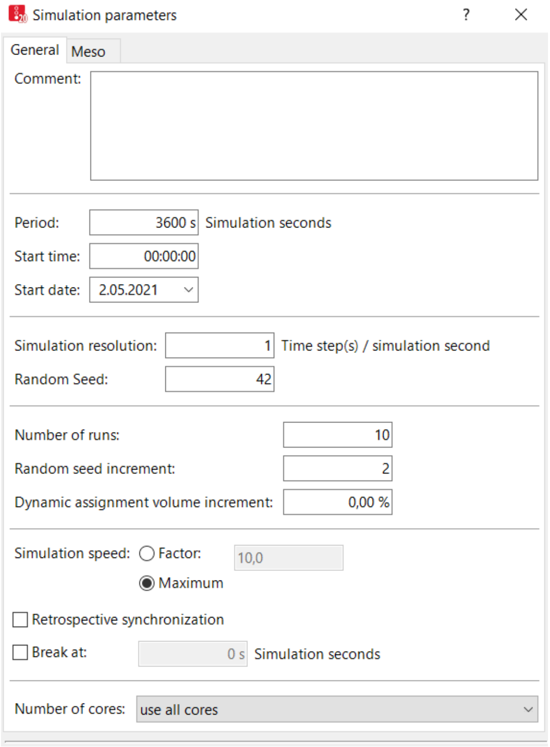


Figure 6.28: PTV VISSIM - Simulation Parameters

After all these steps, simulation and environment parameter values can be seen in table 6.36.

Table 6.36: Simulation and Environment Parameters - Intersection 3

Number Of Lanes	14
Number Of Phases	4
Yellow Time After Actual Green Time	3 seconds
All-Red Time	1 seconds
Vehicle Compositions	100% Passenger Cars
Desired Vehicle Speed	30 km/h
Delay Segment (m)	100,200,300 and 400
W74bxAdd	3.00
W74bxMult	3.00
#Number Of Runs	10
Initial Random Seed Value	42
Random Seed Value Increment	2

6.0.4.5 Delay Comparison Between PTV VISSIM and HCM 2000 Delay Model

For Intersection 3, each scenario was simulated in PTV VISSIM 10 times and the delay results were noted. Accordingly, for each scenario, 1-Hour average control delay of HCM 2000 and 1-hour VISSIM average vehicle delay results can be seen in table 6.37.

Table 6.37: HCM 2000 Delay vs Vissim Delay For Each Scenario - Intersection 3

Scenario No	*1-Hour HCM 2000 Average Vehicle Delay (sec/veh)*	*1-Hour PTV VISSIM Average Vehicle Delay (sec/veh)*	*Delay Difference Percentage (%)*
1	**22.58**	**21.21**	**-%6.0**
2	**24.75**	**24.00**	**-%3.0**
3	**47.12**	**49.50**	**%5.0**
4	**47.40**	**47.42**	**%0.04**
5	**51.27**	**47.43**	**-%7.4**
6	**51.55**	**52.56**	**-%2.0**
7	**56.97**	**54.62**	**-%4.1**
8	**61.69**	**64.54**	**+%4.6**
9	**71.44**	**78.53**	**+%9.9**
10	**92.54**	**95.26**	**+%2.9**
11	**125.90**	**129.12**	**+%2.6**
12	**142.36**	**141.96**	**-%0.3**

As seen from the table, although the delays of VISSIM and HCM 2000 are different in each scenario, this is a predicted result because of the stochastic nature of PTV VISSIM. However, the delay differences are fairly low except for a few scenarios (Scenario 5 and 9). In addition, delay differences in percentage are generally around **4%**. Based on these results, it can be said that the intersection characteristics and lane-based saturation flow rates are successfully reflected in the PTV VISSIM environment.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions and Recommendations

In this thesis, we propose a model comprising multiple objective functions to the traffic signal control problem using linear programming optimization technique. The model has the ability to suggest optimized green times and signal settings for both the under-saturated and the over-saturated conditions.

Two approaches, MCLM and CCM, have been developed for under-saturated conditions. In some traffic scenarios and intersection geometries, MCLM gave lower delay results, while in others, CCM gave lower delay results. This situation is because the MCLM approach leaves high degrees of saturation in some lane groups. A high degree of saturation means vehicle arrivals or flow rates nearly approaching capacity and thus this situation can increase vehicle delays according to HCM 2000 delay formulas. The conclusion that can be drawn from this is that minimizing cycle length as much as possible will not always give the lowest delays. For this reason, it may be useful to use different approaches in under-saturated conditions.

In over-saturated conditions, MTQLM approach, which tries to minimize the total queue length, leaves unfair queues in some lane groups. This situation becomes even worse, especially in high-volume traffic scenarios. As a result, as the analysis time increases, vehicles arriving on those lane groups, to pass through the intersection, must wait for more green periods, and thus this results in high delays in those lane groups. For this reason, the second method, MMQLM, which can be used in over-saturated conditions, treats lane groups more fairly as it takes vehicle arrival rates into account while allocating the total residual queue. Because of this, more fair delays occur in all lane groups. According to the results of the experiments, MMQLM reduced the average vehicle delays of intersection up to 40% compared to the MTQLM approach, especially in some high-volume traffic scenarios. Even if the average vehicle delays obtained using these two approaches are close to each other, MMQLM approach, which gives more balanced delay results in lane groups, may be preferred by decision-makers.

In the experiments, it was concluded that the NSM approach, which is used to further reduce the average vehicle delays at intersections by using the green times obtained from the first scene, always gives lower delay results at every intersection and in every scenario in the over-saturated conditions. Also, because NSM is a neighbor-search-based algorithm, it has a quite fast run time. For this reason, this 2-stage approach, which is proposed for over-saturated conditions, can be used as an alternative to other linear programming models in the literature.

Besides the models developed, the importance of phase design was investigated in this thesis. In the experiments, two possible phase designs were evaluated at the 3rd intersection. It was concluded that using the second phase designs gives less delay and is more advantageous than the first phase design in most scenarios. In some scenarios, the use of the second phase design has reduced average vehicle delays by up to 50% compared to the first phase design. This is because in the first phase design, lane groups with high traffic volume gained right of way in different phases, whereas in the second phase design, lane groups with high traffic volumes gained right of way in the same phases, considering potential conflicts at the intersection. By using the second phase design, lane groups with high traffic volumes were assigned more green time and thus over-saturated conditions have occurred in fewer scenarios. For example, for scenario 8, 9 and 10, at the 3rd intersection, over-saturated conditions occurred using the first phase design, while under-saturated conditions occurred in these scenarios using the second phase design. Since over-saturated conditions at intersections cause high delays, over-saturated conditions should be avoided as much as possible to reduce delays. The conclusion that can be drawn from this is that besides the optimization of the phase durations in the traffic signal control problem, all possible phase designs should also be considered and therefore optimized to reduce delays as much as possible.

7.2 Future Work

Within the scope of this thesis, linear programming models have been developed for optimum fixed-time traffic signal control at isolated signalized intersections. Since fixed-time traffic signal control strategies are suitable for arterial road networks [46], we plan to apply the methods we have developed in coordinated arterial networks in the future. Additionally, the methods we have developed can be further improved and used for optimization in grid road networks comprising multiple intersections.

The fixed-time traffic signal optimization we have developed can not respond to sudden fluctuations in the traffic flow. Therefore, in the future, this fixed-time system can be transformed into an adaptive signal control strategy that can respond to fluctuations in traffic. With the adaptive signal control strategy that can be created by combining linear programming models with the PTV VISSIM VisVap module, we plan to seek solutions for non-recurrent traffic congestion that may result from accidents, sudden bad weather and breakdowns in the future.

In the experiments, it was concluded that optimizing the phase design also has positive effects on vehicle delays. For this reason, optimizing the possible phase designs according to different intersection geometries is among the improvements that can be made in the future.

REFERENCES

[1] L. Singh, S. Tripathi, and H. Arora, "Time Optimization for Traffic Signal Control Using Genetic Algorithm," *International Journal of Recent Trends in Engineering*, vol. 2, no. 2, p. 4, 2009.

[2] D. Teodorović and M. Dell'Orco, "Mitigating traffic congestion: Solving the ride-matching problem by Bee Colony Optimization," *Transportation Planning and Technology*, vol. 31, no. 2, pp. 135–152, 2008.

[3] "İstanbul büyükşehir belediyesinin 2021 bütçesi kabul edildi," November 19, 2020 [Online] Accessed on: May 19 , 2021 Available:. `https://www.aa.com.tr/tr/turkiye/istanbul-buyuksehir-belediyesinin-2021-butcesi-kabul-edildi/2049737`.

[4] "PTV VISSIM 10 USER MANUAL," *PTV AG, Karlsruhe, Germany*, p. 1155, 2018.

[5] M. Eom and B. I. Kim, "The traffic signal control problem for intersections: a review," *European Transport Research Review*, vol. 12, no. 1, 2020.

[6] "HCM (2000) Highway Capacity Manual," *Transport Research Board, USA.*

[7] C. J. Bester and W. L. Meyers, "Saturation flow rates," *SATC 2007 - 26th Annual Southern African Transport Conference: The Challenges of Implementing Policy*, no. July, pp. 560–568, 2007.

[8] N. H. Gartner, C. J. Messer, and A. K. Rathi, "Traffic flow theory - a state-of-the-art report," *TRB Special Report 165 - Revised Monograph on Traffic Flow Theory*, p. Chapter 9, 2001.

[9] F. Dion, H. Rakha, and Y. S. Kang, "Comparison of delay estimates at under-saturated and over-saturated pre-timed signalized intersections," *Transportation Research Part B: Methodological*, vol. 38, no. 2, pp. 99–122, 2004.

[10] A. Koźlak and D. Wach, "Causes of traffic congestion in urban areas. Case of Poland," *SHS Web of Conferences*, vol. 57, 2018.

[11] A. Downs, "Still Stuck in Traffic: Coping with Peak-Hour Traffic Congestion," *Brookings Institution Press*, p. 472, 2004.

[12] G. Weisbrod, D. Vary, and G. Treyz, "Economic Implications of Congestion," *NCHRP REPORT 463,National Cooperative Highway Research Program,Transportation Research Board*, 2001.

[13] A. Skabardonis, P. Varaiya, and K. F. Petty, "Measuring Recurrent and Nonrecurrent Traffic Congestion," *Transportation Research Record 1856*, pp. 118–124, 2003.

[14] J. C. Falcocchio and H. S. Levinson, "Road Traffic Congestion: A Concise Guide," *Springer Tracts on Transportation and Traffic*, 2015.

[15] D. Shinar, "Aggressive driving: The contribution of the drivers and the situation," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 1, no. 2, pp. 137–160, 1998.

[16] X.-F. Xie, S. F. Smith, G. J. Barlow, and T.-W. Chen, "Coping with Real-World Challenges in Real-Time Urban Traffic Control," *Transportation Research Board (TRB) Annual Meeting At: Washington, DC, USA*, 2014.

[17] P. R. Lowre, "SCATS, Sydney Co-Ordinated Adaptive Traffic System : a traffic responsive method of controlling urban traffic," *Roads and Traffic Authority NSW, Darlinghurst, NSW Australia*, 1990.

[18] D. I. Robertson and R. D. Bretherton, "Optimizing Networks of Traffic Signals in Real Time—The SCOOT Method," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 1, pp. 11–15, 1991.

[19] C. E. Wallace, K. G. Courage, M. A. Hadi, and A. C. Gan, "TRANSYT-7F user's guide," *Transportation Research Center,University of Florida, Gainesville*, 1998.

[20] P. Mirchandani and L. Head, “A real-time traffic signal control system: Architecture, algorithms, and analysis,” *Transportation Research Part C: Emerging Technologies*, vol. 9, no. 6, pp. 415–432, 2001.

[21] D. Husch and J. Albeck, “Synchro 6: Traffic Signal Software - User Guide,” *Trafficware, Limited*, 2003.

[22] D. Srinivasan, M. C. Choy, and R. L. Cheu, “Neural networks for real-time traffic signal control,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 3, pp. 261–272, 2006.

[23] G. B. Castro, A. R. Hirakawa, and J. S. Martini, “Adaptive traffic signal control based on bio-neural network,” *Procedia Computer Science*, vol. 109, no. 2016, pp. 1182–1187, 2017.

[24] M. B. Trabia, M. S. Kaseko, and M. Ande, “A two-stage fuzzy logic controller for traffic signals,” *Transportation Research Part C: Emerging Technologies*, vol. 7, no. 6, pp. 353–367, 1999.

[25] S. Chiu, “Adaptive traffic signal control using fuzzy logic,” *In: Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 98–1007, 1992.

[26] J. Zeng, J. Hu, and Y. Zhang, “Adaptive Traffic Signal Control with Deep Recurrent Q-learning,” *IEEE Intelligent Vehicles Symposium, Proceedings*, vol. 2018-June, no. Iv, pp. 1215–1220, 2018.

[27] X. Liang, X. Du, G. Wang, and Z. Han, “Deep reinforcement learning for traffic light control in vehicular networks,” *IEEE Transactions On Vehicular Technology*, vol. XX, no. Xx, pp. 1–11, 2018.

[28] L. Shoufeng, L. Ximin, and D. Shiqiang, “Q-Learning for Adaptive Traffic Signal Control Based on Delay Minimization Strategy,” *IEEE International Conference on Networking, Sensing and Control*, pp. 687–691, 2008.

[29] E. Van der Pol and F. A. Oliehoek, “Coordinated deep reinforcement learners for traffic light control,” *30th Conference on Neural Information Processing Systems*, no. Nips, p. 8, 2016.

[30] Y. M. Çetinkaya, Z. S. Doğu, and O. Dayıbaş, "On the development of the autonomous city," in *5th International Symposium on Innovative Technologies in Engineering and Science 29-30 September 2017 (ISITES2017 Baku-Azerbaijan)*, 2017.

[31] P. Srisurin and A. Singh, "Optimal Signal Plan for Minimizing Queue Lengths at a Congested Intersection," *International Journal of Traffic and Transportation Engineering*, vol. 6, no. 3, pp. 53–63, 2017.

[32] H. Liu, K. N. Balke, and W. H. Lin, "A reverse causal-effect modeling approach for signal control of an oversaturated intersection," *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 6, pp. 742–754, 2008.

[33] P. Coll, P. Factorovich, I. Loiseau, and R. Gómez, "A linear programming approach for adaptive synchronization of traffic signals," *International Transactions in Operational Research*, vol. 20, no. 5, pp. 667–679, 2013.

[34] B. Park, C. J. Messer, and T. Urbanik II, "Traffic Signal Optimization Program for Oversaturated Conditions: Genetic Algorithm Approach," *Transportation Research Record*, vol. 1683, pp. 133–142, 1999.

[35] Y. K. Chin, K. C. Yong, N. Bolong, S. S. Yang, and K. T. Teo, "Multiple intersections traffic signal timing optimization with genetic algorithm," *In Proceeding of International Conference on Control System, Computing and Engineering, ICCSCE 2011*, pp. 454–459, 2011.

[36] A. Stevanovic, P. T. Martin, and J. Stevanovic, "VisSim-based genetic algorithm optimization of signal timings," *Transportation Research Record*, no. 2035, pp. 59–68, 2007.

[37] M. A. S. Kamal, J. ichi Imura, T. Hayakawa, A. Ohata, and K. Aihara, "Traffic Signal Control of a Road Network Using MILP in the MPC Framework," *International Journal of Intelligent Transportation Systems Research*, vol. 13, no. 2, pp. 107–118, 2015.

[38] H. Y. Sutarto, M. Maulida, E. Joelianto, and A. Samsi, "Queue length optimization of vehicles at road intersection using parabolic interpolation method," *Proceedings of the 2015 International Conference on Automation, Cognitive*

Science, Optics, Micro Electro-Mechanical System, and Information Technology, ICACOMIT 2015, pp. 63–67, 2016.

[39] S. A. Celtek, A. Durdu, and M. E. M. Alı, "Real-time traffic signal control with swarm optimization methods," *Measurement: Journal of the International Measurement Confederation*, vol. 166, p. 108206, 2020.

[40] Z. ÇAKICI, "Sinyalize Kavşaklar İçin Optimizasyon Tabanlı Trafik Yönetim Modeli," *Pamukkale Üniversitesi Fen Bilimleri Enstitüsü İnşaat Mühendisliği Anabilim Dalı*, p. 173, PhD Thesis, February 2020.

[41] H. Ozen and E. N. Unlu, "Sinyal Faz Diyagramının Kavşak Performansı Üzerindeki Etkisinin İncelenmesi : Antalya Örneği," pp. 309–318.

[42] P. Fogliaroni, D. Bucher, N. Jankovic, and I. Giannopoulos, "Intersections of our world," *Leibniz International Proceedings in Informatics, LIPIcs*, vol. 114, pp. 0–15, 2018.

[43] X. Kong, G. Shen, F. Xia, and C. Lin, "Urban arterial traffic two-direction green wave intelligent coordination control technique and its application," *International Journal of Control, Automation and Systems*, vol. 9, no. 1, pp. 60–68, 2011.

[44] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, "Review of road traffic control strategies," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2043–2065, 2003.

[45] F. JAVANSHOUR, "Delay-Based Performance Analyses of Four-Legged Signalized Intersections: A Case Study," *Istanbul Technical University, Graduate School of Science Engineering and Technology*, p. 135, Msc Thesis, May 2014.

[46] "Signal Timing Manual," *Department of Transportation, Federal highway administration FHWA*, p. 274, 2008.

[47] O. Erarslan, "IŞIKLI KAVŞAKLARDA AMERİKAN VE AVUSTRALYA YÖNTEMLERİ İLE GECİKME ANALİZİ VE ÖRNEK BİR KAVŞAK ÇÖZÜMÜ," *İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü*, p. 173, Msc Thesis, October 2008.

[48] F. Awadallah, "Yellow and All-Red Intervals: How To Improve Safety and Reduce Delay?," *International Journal for Traffic and Transport Engineering*, vol. 3, no. 2, pp. 159–172, 2013.

[49] C. Q. Shao, J. Rong, and X. M. Liu, "Study on the saturation flow rate and its influence factors at signalized intersections in China," *Procedia - Social and Behavioral Sciences*, vol. 16, pp. 504–514, 2011.

[50] K. Hamad and H. Abuhamda, "Estimating Base Saturation Flow Rate for Selected Signalized Intersections in Doha, Qatar," *Journal of Traffic and Logistics Engineering*, vol. 3, no. 2, 2015.

[51] S. Dündar and K. S. Öğüt, "Determination of Basic Saturation Flow Rate in Istanbul," *International Journal of Engineering Technologies IJET*, vol. 4, no. 1, pp. 47–52, 2018.

[52] J. Asamer and H. J. Van Zuylen, "Saturation flow under adverse weather conditions," *Transportation Research Record*, no. 2258, pp. 103–109, 2011.

[53] L. Jie, H. J. Van Zuylen, Y. S. Chen, and R. Lu, "Comparison of driver behaviour and saturation flow in China and the Netherlands," *IET Intelligent Transport Systems*, vol. 6, no. 3, pp. 318–327, 2012.

[54] F. V. Webster, "Traffic Signal Settings Report," *Michigan University: H.M. Stationery Office*, 1958.

[55] R. Akçelik, "Traffic Signals: Capacity and Timing Analysis," *Australian Road Research Board*, 1981.

[56] Y. S. Murat, "Sinyalize Kavşaklardaki Taşıt Gecikmelerinin Bulanık Mantık ile Modellenmesi," *İMO Teknik Dergi*, vol. 17, pp. 3903–3916, 2006.

[57] S. S. S. M. Qadri, M. A. Gökçe, and E. Öner, "State-of-art review of traffic signal control methods: challenges and opportunities," *European Transport Research Review*, vol. 12, no. 1, pp. 1–23, 2020.

[58] M. Fellendorf and P. Vortisch, "Microscopic traffic flow simulator VISSIM," *International Series in Operations Research & Management Science*, vol. 145, pp. 63–93, 2010.

[59] R. Wiedemann, "SIMULATION DES STRASSENVERKEHRSFLUSSES. ," *Schriftenreihe des Instituts für Verkehrswesen der Universität Karlsruhe*, 1974.

[60] Z. Lu, T. Fu, L. Fu, S. Shiravi, and C. Jiang, "A video-based approach to calibrating car-following parameters in VISSIM for urban traffic," *International Journal of Transportation Science and Technology*, vol. 5, no. 1, pp. 1–9, 2016.

[61] G. Cornuejols and R. Tütüncü, "Optimization Methods in Finance," *Cambridge University Press, Cambridge*, 2006.

[62] N. A. Sofi, A. Ahmed, M. Ahmad, and B. A. Bhat, "Decision Making in Agriculture: A Linear Programming Approach," *International Journal of Modern Mathematical Sciences Journal*, vol. 13, no. 2, pp. 160–169, 2015.

[63] J. C. T. Mao, "Application of Linear Programming to Short-Term Financing Decision," *The Engineering Economist*, vol. 13, no. 4, pp. 221–241, 1968.

[64] L. H. Shih, "Cement transportation planning via fuzzy linear programming," *International Journal of Production Economics*, vol. 58, no. 3, pp. 277–287, 1999.

[65] E. K. P. Chong and S. H. Zak, "An Introduction to Optimization," *A Wiley-lnterscience Publication*, p. 495, 2001.

[66] F. Hiller and G. J. C. T. Lieberman, "Introduction to Operation Research," *McGraw Hill, New York.*, p. 1237, 1990.

[67] "Linear programming: Advantages, disadvantages and strategies," November , 2018 [Online] Accessed on: May 16 , 2021 Available:. `https://www.ukessays.com/essays/management/i-linear-programming.php?vref=1`.

[68] H. WEI, G. ZHENG, V. GAYAH, and Z. LI, "A survey on traffic signal control methods," *arXiv*, vol. 1, no. 1, 2019.

APPENDIX A

NUMERICAL FORMULATIONS OF LP MODELS

Table A.1: Scenario 1 - MTQLM - Numerical Linear Formulization

Decision Variables	*Phase_1, Phase_2, Phase_3, Phase_4*
Objective Function	*(Minimize) ((0.54 * 135 - 3 * 0.5 * Phase_1) + ... + (0.152 * 134 - 1 * 0.5 * Phase_3))*
Discharge Constraint	*(Discharging) Phase_1 * 0.5 * 3 <= 135 * 0.54 (Cum. Arriving)*
	*(Discharging) Phase_2 * 0.5 * 1 <= 135 * 0.083 (Cum. Arriving)*
	*(Discharging) Phase_3 * 0.5 * 1 <= 135 * 0.152 (Cum. Arriving)*
	*(Discharging) Phase_4 * 0.5 * 1 <= 135 * 0.125 (Cum. Arriving)*
Min. Green Constraint	*Phase_1 >= 9*
	Phase_2 >= 9
	Phase_3 >= 9
	Phase_4 >= 9
Cycle Constraint	*Phase_1 + Phase_2 + Phase_3 + Phase_4 == 123*

Figure A.1: Scenario 1 - MTQLM - Pulp Package Output

```
Traffic_Light_Optimization:
MINIMIZE
-2.5*Phase_1 + -1.0*Phase_2 + -0.5*Phase_3 + -0.5*Phase_4 + 151.875
SUBJECT TO
_C1: Phase_1 >= 9

_C2: Phase_2 >= 9

_C3: Phase_3 >= 9

_C4: Phase_4 >= 9

_C5: 1.5 Phase_1 <= 72.9

_C6: 0.5 Phase_2 <= 11.25

_C7: 0.5 Phase_3 <= 20.625

_C8: 0.5 Phase_4 <= 16.875

_C9: Phase_1 + Phase_2 + Phase_3 + Phase_4 = 123

VARIABLES
Phase_1 free Integer
Phase_2 free Integer
Phase_3 free Integer
Phase_4 free Integer

Maximum Value :  -16.625
Optimal
Phase_1:  48.0 Phase_2:  22.0 Phase_3:  20.0 Phase_4 33.0 Toplam Işık Süresi : 135.0
```

Table A.2: Scenario 1 - MMQLM - Numerical Linear Formulization

Decision Variables	*Phase_1, Phase_2, Phase_3, Phase_4*
Objective Function	*minimize (max_queue)*
	*max_queue >= (0.54 * 135 - 3 * 0.5 * Phase_1 / 0.6)*
	*max_queue >= (0.083 * 135 - 1 * 0.5 * Phase_2 / 0.092)*
	*max_queue >= (0.152 * 135 - 1 * 0.5 * Phase_3 / 0.170)*
	*max_queue >= (0.125 * 135 - 1 * 0.5 * Phase_4 / 0.139)*
Discharge Constraint	*(Discharging) Phase_1 * 0.5 * 3 <= 135 * 0.54 (Cum. Arriving)*
	*(Discharging) Phase_2 * 0.5 * 1 <= 135 * 0.083 (Cum. Arriving)*
	*(Discharging) Phase_3 * 0.5 * 1 <= 135 * 0.152 (Cum. Arriving)*
	*(Discharging) Phase_4 * 0.5 * 1 <= 135 * 0.125 (Cum. Arriving)*
Min. Green Constraint	*Phase_1 >= 9*
	Phase_2 >= 9
	Phase_3 >= 9
	Phase_4 >= 9
Cycle Constraint	*Phase_1 + Phase_2 + Phase_3 + Phase_4 == 123*

Figure A.2: Scenario 1 - MMQLM - Pulp Package Output

```
Traffic_Light_Optimization:
MINIMIZE
1*Maximum + 0
SUBJECT TO
_C1: Maximum + 2.50308641975 Phase_1 >= 121.65

_C2: Maximum + 5.40666666667 Phase_2 >= 121.65

_C3: Maximum + 2.94909090909 Phase_3 >= 121.65

_C4: Maximum + 3.60444444444 Phase_4 >= 121.65

_C5: Phase_1 >= 9

_C6: Phase_2 >= 9

_C7: Phase_3 >= 9

_C8: Phase_4 >= 9

_C9: 1.5 Phase_1 <= 72.9

_C10: 0.5 Phase_2 <= 11.25

_C11: 0.5 Phase_3 <= 20.625

_C12: 0.5 Phase_4 <= 16.875

_C13: Phase_1 + Phase_2 + Phase_3 + Phase_4 = 123

VARIABLES
Maximum free Integer
Phase_1 free Integer
Phase_2 free Integer
Phase_3 free Integer
Phase_4 free Integer

Maximum Value :  21.0
Optimal
Phase_1:  41.0 Phase_2:  19.0 Phase_3:  35.0 Phase_4 28.0 Toplam Işık Süresi : 135.0
```

Table A.3: Scenario 3 - MCLM - Numerical Linear Formulization

Decision Variables	*Phase_1, Phase_2, Phase_3, Phase_4*
Objective Function	*(minimize) (Phase_1 + Phase_2 + Phase_3 + Phase_4 + 12)*
Discharge Constraint	*(Cum. Arriving) (Phase_1 + ... + Phase_4 + 12) * 0.24 <= Phase_1 * 3 * 0.5 (Discharging)*
	*(Cum. Arriving) (Phase_1 + ... + Phase_4 + 12) * 0.05 <= Phase_2 * 1 * 0.5 (Discharging)*
	*(Cum. Arriving) (Phase_1 + ... + Phase_4 + 12) * 0.08 <= Phase_3 * 1 * 0.5 (Discharging)*
	*(Cum. Arriving) (Phase_1 + ... + Phase_4 + 12) * 0.05 <= Phase_4 * 1 * 0.5 (Discharging)*
Min. Green Constraint	*Phase_1 >= 9*
	Phase_2 >= 9
	Phase_3 >= 9
	Phase_4 >= 9

Figure A.3: Scenario 3 - MCLM - Pulp Package Output

```
Traffic_Light_Optimization:
MINIMIZE
1*Phase_1 + 1*Phase_2 + 1*Phase_3 + 1*Phase_4 + 12
SUBJECT TO
_C1: Phase_1 >= 9

_C2: Phase_2 >= 9

_C3: Phase_3 >= 9

_C4: Phase_4 >= 9

_C5: 1.26 Phase_1 - 0.24 Phase_2 - 0.24 Phase_3 - 0.24 Phase_4 >= 2.88

_C6: - 0.05 Phase_1 + 0.45 Phase_2 - 0.05 Phase_3 - 0.05 Phase_4 >= 0.6

_C7: - 0.08 Phase_1 - 0.08 Phase_2 + 0.42 Phase_3 - 0.08 Phase_4 >= 0.96

_C8: - 0.05 Phase_1 - 0.05 Phase_2 - 0.05 Phase_3 + 0.45 Phase_4 >= 0.6

_C9: 0 Phase_1 + 0 Phase_2 + 0 Phase_3 + 0 Phase_4 = 0

VARIABLES
Phase_1 free Integer
Phase_2 free Integer
Phase_3 free Integer
Phase_4 free Integer

Maximum Value :  48.0
Optimal
Phase_1:  9.0 Phase_2:  9.0 Phase_3:  9.0 Phase_4 9.0 Toplam Işık Süresi : 48.0
```

Table A.4: Scenario 3 - CCM - Numerical Linear Formulization

Decision Variables	*Phase_1, Phase_2, Phase_3, Phase_4*
Objective Function	*(minimize) ((135 * 0.24 - 3 * 0.5 * Phase_1) + ... + (0.05 * 135 - 1 * 0.5 * Phase_3))*
Discharge Constraint	*(Discharging) Phase_1 * 3 * 0.5 <= 1.75 * 135 * 0.24 (Cum. Arriving)*
	*(Discharging) Phase_2 * 1 * 0.5 <= 1.75 * 135 * 0.05 (Cum. Arriving)*
	*(Discharging) Phase_3 * 1 * 0.5 <= 1.75 * 135 * 0.08 (Cum. Arriving)*
	*(Discharging) Phase_4 * 1 * 0.5 <= 1.75 * 135 * 0.05 (Cum. Arriving)*
Min. Green Constraint	*Phase_1 >= 9*
	Phase_2 >= 9
	Phase_3 >= 9
	Phase_4 >= 9
Cycle Constraint	*Phase_1 + Phase_2 + Phase_3 + Phase_4 == 123*

Figure A.4: Scenario 3 - CCM - Pulp Package Output

```
Traffic_Light_Optimization:
MINIMIZE
-2.5*Phase_1 + -1.0*Phase_2 + -0.5*Phase_3 + -0.5*Phase_4 + 70.65
SUBJECT TO
_C1: Phase_1 >= 9

_C2: Phase_2 >= 9

_C3: Phase_3 >= 9

_C4: Phase_4 >= 9

_C5: 1.5 Phase_1 <= 57

_C6: 0.5 Phase_2 <= 12

_C7: 0.5 Phase_3 <= 19

_C8: 0.5 Phase_4 <= 12

_C9: Phase_1 + Phase_2 + Phase_3 + Phase_4 = 123

VARIABLES
Phase_1 free Integer
Phase_2 free Integer
Phase_3 free Integer
Phase_4 free Integer

Maximum Value :  -78.85
Optimal
Phase_1:  38.0 Phase_2:  24.0 Phase_3:  37.0 Phase_4 24.0 Toplam Işık Süresi : 135.0
```